Lecture notes on ridge regression

Version 0.20, August 23, 2018.

Wessel N. van Wieringen[1,2]

[1] Department of Epidemiology and Biostatistics, VU University Medical Center
P.O. Box 7057, 1007 MB Amsterdam, The Netherlands
[2] Department of Mathematics, VU University Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: w.vanwieringen@vumc.nl

Disclaimer

This document is a collection of many well-known results on ridge regression. The current status of the document is 'work-in-progress' as it is incomplete (more results from literature will be included) and it may contain inconsistencies and errors. Hence, reading and believing at own risk. Finally, proper reference to the original source may sometimes be lacking. This is regrettable and these references (if ever known to the author) will be included in later versions.

Contents

1   Ridge regression

High-throughput techniques measure many characteristics of a single sample simultaneously. The number of characteristics p measured may easily exceed ten thousand. In most medical studies the number of samples n involved often falls behind the number of characteristics measured, i.e: $p > n$. The resulting $(n \times p)$-dimensional data matrix $\mathbf{X}$:

$$\mathbf{X} = (X_{*,1} \mid \ldots \mid X_{*,p}) = \begin{pmatrix} X_{1,*} \\ \vdots \\ X_{n,*} \end{pmatrix} = \begin{pmatrix} X_{1,1} & \ldots & X_{1,p} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \ldots & X_{n,p} \end{pmatrix}$$

from such a study contains a larger number of covariates than samples. When $p > n$ the data matrix $\mathbf{X}$ is said to be *high-dimensional*.

In this chapter we adopt the traditional statistical notation of the data matrix. An alternative notation would be $\mathbf{X}^\top$ (rather than $\mathbf{X}$), which is employed in the field of (statistical) bioinformatics. In $\mathbf{X}^\top$ the rows comprise the samples rather than the covariates. The case for the bioinformatics notation stems from practical arguments. A spreadsheet is designed to have more rows than columns. In case $p > n$ the traditional notation yields a spreadsheet with more columns than rows. When $p > 10000$ the conventional display is impractical. In this chapter we stick to the conventional statistical notation of the data matrix as all mathematical expressions involving $\mathbf{X}$ are then in line with those of standard textbooks on regression.

The information contained in $\mathbf{X}$ is often used to explain a particular property of the samples involved. In applications in molecular biology $\mathbf{X}$ may contain microRNA expression data from which the expression levels of a gene are to be described. When the gene's expression levels are denoted by $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$, the aim is to find the linear relation $Y_i = \mathbf{X}_{i,*} \boldsymbol{\beta}$ from the data at hand by means of regression analysis. Regression is however frustrated by the high-dimensionality of $\mathbf{X}$ (illustrated in Section 1.2 and at the end of Section 1.5). These notes discuss how regression may be modified to accommodate the high-dimensionality of $\mathbf{X}$. First, however, 'standard' linear regression is recaputilated.

1.1   Linear regression

Consider an experiment in which p characteristics of n samples are measured. The data from this experiment are denoted $\mathbf{X}$, with $\mathbf{X}$ as above. The matrix $\mathbf{X}$ is called the *design matrix*. Additional information of the samples is available in the form of $\mathbf{Y}$ (also as above). The variable $\mathbf{Y}$ is generally referred to as the *response variable*. The aim of regression analysis is to explain $\mathbf{Y}$ in terms of $\mathbf{X}$ through a functional relationship like $Y_i = f(\mathbf{X}_{i,*})$. When no prior knowledge on the form of $f(\cdot)$ is available, it is common to assume a linear relationship between $\mathbf{X}$ and $\mathbf{Y}$. This assumption gives rise to the *linear regression model*:

$$\begin{aligned} Y_i &= \mathbf{X}_{i,*} \boldsymbol{\beta} + \varepsilon_i \\ &= \beta_1 X_{i,1} + \ldots + \beta_p X_{i,p} + \varepsilon_i. \end{aligned} \tag{1.1}$$

In model (1.1) $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ is the *regression parameter*. The parameter β_j, $j = 1, \ldots, p$, represents the effect size of covariate j on the response. That is, for each unit change in covariate j (while keeping the other covariates fixed) the observed change in the response is equal to β_j. The second summand on the right-hand side of the model, ε_i, is referred to as the error. It represents the part of the response not explained by the functional part $\mathbf{X}_{i,*} \boldsymbol{\beta}$ of the model (1.1). In contrast to the functional part, which is considered to be systematic (i.e. non-random), the error is assumed to be random. Consequently,

$Y_{i_1,*}$ need not equal $Y_{i_2,*}$ for $i_1 \neq i_2$, even if $\mathbf{X}_{i_1,*} = \mathbf{X}_{i_2,*}$. To complete the formulation of model (1.1) we need to specify the probability distribution of ε_i. It is assumed that $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and the ε_i are independent, i.e.:

$$\mathrm{Cov}(\varepsilon_{i_1}, \varepsilon_{i_2}) \;\; = \;\; \left\{ \begin{array}{ll} \sigma^2 & \text{if} \;\; i_1 = i_2, \\ 0 & \text{if} \;\; i_1 \neq i_2. \end{array} \right.$$

The randomness of ε_i implies that $\mathbf{Y}_i$ is also a random variable. In particular, $\mathbf{Y}_i$ is normally distributed, because $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\mathbf{X}_{i,*}\boldsymbol{\beta}$ is a non-random scalar. To specify the parameters of the distribution of $\mathbf{Y}_i$ we need to calculate its first two moments. Its expectation equals:

$$\mathbb{E}(Y_i) \;\; = \;\; \mathbb{E}(\mathbf{X}_{i,*}\boldsymbol{\beta}) + \mathbb{E}(\varepsilon_i) \;\; = \;\; \mathbf{X}_{i,*}\boldsymbol{\beta},$$

while its variance is:

$$\begin{aligned} \mathrm{Var}(Y_i) \;\; &= \;\; \mathbb{E}\{[Y_i - \mathbb{E}(Y_i)]^2\} \;\; = \;\; \mathbb{E}(Y_i^2) - [\mathbb{E}(Y_i)]^2 \\ &= \;\; \mathbb{E}[(\mathbf{X}_{i,*}\boldsymbol{\beta})^2 + 2\varepsilon_i\mathbf{X}_{i,*}\boldsymbol{\beta} + \varepsilon_i^2] - (\mathbf{X}_{i,*}\boldsymbol{\beta})^2 \\ &= \;\; \mathbb{E}(\varepsilon_i^2) \;\; = \;\; \mathrm{Var}(\varepsilon_i) \;\; = \;\; \sigma^2. \end{aligned}$$

Hence, $Y_i \sim \mathcal{N}(\mathbf{X}_{i,*}\boldsymbol{\beta}, \sigma^2)$. This formulation (in terms of the normal distribution) is equivalent to the formulation of model (1.1), as both capture the assumptions involved: the linearity of the functional part and the normality of the error.

Model (1.1) is often written in a more condensed matrix form:

$$\mathbf{Y} \;\; = \;\; \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.2}$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^\top$ and distributed as $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_p, \sigma^2 \mathbf{I}_{nn})$. As above model (1.2) can be expressed as a multivariate normal distribution: $\mathbf{Y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_{nn})$.

Model (1.2) is a so-called hierarchical model. This terminology emphasizes that $\mathbf{X}$ and $\mathbf{Y}$ are not on a par, they play different roles in the model. The former is used to explain the latter. In model (1.1) $\mathbf{X}$ is referred as the *explanatory* or *independent* variable, while the variable $\mathbf{Y}$ is generally referred to as the *response* or *dependent* variable.

The covariates, the columns of $\mathbf{X}$, may themselves be random. To apply the linear model they are temporarily assumed fixed. The linear regression model is then to be interpreted as $\mathbf{Y} \,|\, \mathbf{X} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_{nn})$

Example 1.1 *Methylation of a tumor-suppressor gene*
Consider a study which measures the gene expression levels of a tumor-suppressor genes (TSG) and two methylation markers (MM1 and MM2) on 67 samples. A methylation marker is a gene that promotes methylation. Methylation refers to attachment of a methyl group to a nucleotide of the DNA. In case this attachment takes place in or close by the promotor region of a gene, this complicates the transcription of the gene. Methylation may down-regulate a gene. This mechanism also works in the reverse direction: removal of methyl groups may up-regulate a gene. A tumor-suppressor gene is a gene that halts the progression of the cell towards a cancerous state.

The medical question associated with these data: do the expression levels methylation markers affect the expression levels of the tumor-suppressor gene? To answer this question we may formulate the following linear regression model:

$$Y_{i,\texttt{tsg}} \;\; = \;\; \beta_0 + \beta_{\texttt{mm1}} X_{i,\texttt{mm1}} + \beta_{\texttt{mm2}} X_{i,\texttt{mm2}} + \varepsilon_i,$$

with $i = 1, \ldots, 67$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. The interest focusses on $\beta_{\texttt{mm1}}$ and $\beta_{\texttt{mm2}}$. A non-zero value of at least one of these two regression parameters indicates that there is a linear association between the expression levels of the tumor-suppressor gene and that of the methylation markers.

Prior knowledge from biology suggests that the $\beta_{\texttt{mm1}}$ and $\beta_{\texttt{mm2}}$ are both non-positive. High expression levels of the methylation markers lead to hyper-methylation, in turn inhibiting the transcription of the tumor-suppressor gene. Vice versa, low expression levels of MM1 and MM2 are (via hypo-methylation) associated with high expression levels of TSG. Hence, a negative concordant effect between MM1 and MM2 (on one side) and TSG (on the other) is expected. Of course, the methylation markers may affect expression levels of other genes that in turn regulate the tumor-suppressor gene. The regression parameters $\beta_{\texttt{mm1}}$ and $\beta_{\texttt{mm2}}$ then reflect the indirect effect of the methylation markers on the expression levels of the tumor suppressor gene. $\square$

The linear regression model (1.1) involves the unknown parameters: $\boldsymbol{\beta}$ and σ^2, which need to be learned from the data. The parameters of the regression model, $\boldsymbol{\beta}$ and σ^2 are estimated by means of likelihood maximization. Recall that $Y_i \sim \mathcal{N}(\mathbf{X}_{i,*}\boldsymbol{\beta}, \sigma^2)$ with corresponding density: $f_{Y_i}(y_i) = (2\pi\sigma^2)^{-1/2} \exp[-(y_i - \mathbf{X}_{i*}\boldsymbol{\beta})^2/2\sigma^2]$. The likelihood thus is:

$$L(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\,\sigma} \exp[-(Y_i - \mathbf{X}_{i,*}\boldsymbol{\beta})^2/2\sigma^2],$$

in which the independence of the observations has been used. Because of the concavity of the logarithm, the maximization of the likelihood coincides with the maximum of the logarithm of the likelihood (called the log-likelihood). Hence, to obtain maximum likelihood (ML) estimates of the parameter it is equivalent to find the maximum of the log-likelihood. The log-likelihood is:

$$\mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \sigma^2) = \log[L(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \sigma^2)] = -n\log(\sqrt{2\pi}\,\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{X}_{i,*}\boldsymbol{\beta})^2.$$

After noting that $\sum_{i=1}^{n}(Y_i - \mathbf{X}_{i,*}\boldsymbol{\beta})^2 = \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$, the log-likelihood can be written as:

$$\mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \sigma^2) = -n\log(\sqrt{2\pi}\,\sigma) - \frac{1}{2\sigma^2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2.$$

In order to find the maximum of the log-likelihood, take its derivate with respect to $\boldsymbol{\beta}$:

$$\frac{\partial}{\partial\boldsymbol{\beta}}\mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \sigma^2) = -\frac{1}{2\sigma^2}\frac{\partial}{\partial\boldsymbol{\beta}}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = \frac{1}{\sigma^2}\mathbf{X}^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

Equate this derivative to zero gives the estimating equation for $\boldsymbol{\beta}$:

$$\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{Y}. \tag{1.3}$$

Equation (1.3) is called to the *normal equation*. Pre-multiplication of both sides of the normal equation by $(\mathbf{X}^\top\mathbf{X})^{-1}$ now yields the ML estimator of the regression parameter: $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$, in which it is assumed that $(\mathbf{X}^\top\mathbf{X})^{-1}$ is well-defined.

Along the same lines one obtains the ML estimator of the residual variance. Take the partial derivative of the log-likelihood with respect to σ^2:

$$\frac{\partial}{\partial\sigma}\mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \sigma^2) = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2.$$

Equate the right-hand side to zero and solve for σ^2 to find $\hat{\sigma}^2 = \frac{1}{n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$. In this expression $\boldsymbol{\beta}$ is unknown and the ML estimate of $\boldsymbol{\beta}$ is plugged-in.

With explicit expressions of the ML estimators at hand, we can study their properties. The expectation of the ML estimator of the regression parameter $\boldsymbol{\beta}$ is:

$$\mathbb{E}(\hat{\boldsymbol{\beta}}) = \mathbb{E}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}] = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbb{E}[\mathbf{Y}] = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}.$$

Hence, the ML estimator of the regression coefficients is unbiased.

The variance of the ML estimator of $\boldsymbol{\beta}$ is:

$$\begin{aligned}
\text{Var}(\hat{\boldsymbol{\beta}}) &= \mathbb{E}\{[\hat{\boldsymbol{\beta}} - \mathbb{E}(\hat{\boldsymbol{\beta}})][\hat{\boldsymbol{\beta}} - \mathbb{E}(\hat{\boldsymbol{\beta}})]^\top\} \\
&= \mathbb{E}\{[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} - \boldsymbol{\beta}][(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} - \boldsymbol{\beta}]^\top\} \\
&= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\,\mathbb{E}\{\mathbf{Y}\mathbf{Y}^\top\}\mathbf{X}\,(\mathbf{X}^\top\mathbf{X})^{-1} - \boldsymbol{\beta}\boldsymbol{\beta}^\top \\
&= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\{\mathbf{X}\boldsymbol{\beta}\boldsymbol{\beta}^\top\mathbf{X}^\top + \boldsymbol{\Sigma}\}\mathbf{X}\,(\mathbf{X}^\top\mathbf{X})^{-1} - \boldsymbol{\beta}\boldsymbol{\beta}^\top \\
&= \boldsymbol{\beta}\boldsymbol{\beta}^\top + \sigma^2\,(\mathbf{X}^\top\mathbf{X})^{-1} - \boldsymbol{\beta}\boldsymbol{\beta}^\top = \sigma^2\,(\mathbf{X}^\top\mathbf{X})^{-1},
\end{aligned}$$

in which we have used that $\mathbb{E}(\mathbf{Y}\mathbf{Y}^\top) = \mathbf{X}\boldsymbol{\beta}\boldsymbol{\beta}^\top\mathbf{X}^\top + \sigma^2\,\mathbf{I}_{nn}$. From $\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2\,(\mathbf{X}^\top\mathbf{X})^{-1}$, one obtains an estimate of the variance of the estimate of the j-th regression coefficient: $\hat{\sigma}^2(\hat{\beta}_j) = \hat{\sigma}^2\sqrt{[(\mathbf{X}^\top\mathbf{X})^{-1}]_{jj}}$.

This may be used to construct a confidence interval for the estimates or test the hypothesis $H_0 : \beta_j = 0$. In the latter $\hat{\sigma}^2$ should not be the maximum likelihood estimator, as it is biased. It is then to be replaced by the residual sum-of-squares divided by $n - p$ rather than n.

The prediction of Y_i, denoted $\widehat{Y}_i$, is the expected value of Y_i according the linear regression model (with its parameters replaced by their estimates). The prediction of Y_i thus equals $\mathbb{E}(Y_i; \hat{\boldsymbol{\beta}}, \hat{\sigma}^2) = \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}$. In matrix notation the prediction is:

$$\widehat{\mathbf{Y}} \;=\; \mathbf{X}\hat{\boldsymbol{\beta}} \;=\; \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} \;:=\; \mathbf{H}\mathbf{Y},$$

where $\mathbf{H}$ is the *hat matrix*, as it 'puts the hat' on $\mathbf{Y}$. Note that the hat matrix is a projection matrix, i.e. $\mathbf{H}^2 = \mathbf{H}$ for

$$\mathbf{H}^2 \;=\; \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top \;=\; \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top.$$

Thus, the prediction $\widehat{\mathbf{Y}}$ is an orthogonal projection of $\mathbf{Y}$ onto the space spanned by the columns of $\mathbf{X}$.
With $\hat{\boldsymbol{\beta}}$ available, an estimate of the errors $\hat{\varepsilon}_i$, dubbed the *residuals* are obtained via:

$$\hat{\varepsilon} \;=\; \mathbf{Y} - \widehat{\mathbf{Y}} \;=\; \mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}} \;=\; \mathbf{Y} - \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} \;=\; [\mathbf{I} - \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top]\,\mathbf{Y}.$$

Thus, the residuals are a projection of $\mathbf{Y}$ onto the orthogonal complement of the space spanned by the columns of $\mathbf{X}$. The residuals are to be used in diagnostics, e.g. checking of the normality assumption by means of a normal probability plot.

For more on the linear regression model confer the monograph of Draper and Smith (1998).

1.2 Ridge regression

When the design matrix is high-dimensional, the covariates (the columns of $\mathbf{X}$) are super-collinear. Recall *collinearity* in regression analysis refers to the event of two (or multiple) covariates being highly linearly related. Consequently, the subspace spanned by collinear covariates may not be (or close to not being) of full rank. When the subspace (onto which $\mathbf{Y}$ is projected) is (close to) rank deficient, it is (almost) impossible to separate the contribution of the individual covariates. The uncertainty with respect to the covariate responsible for the variation explained in $\mathbf{Y}$ is often reflected in the fit of the linear regression model to data by a large error of the estimates of the regression parameters corresponding to the collinear covariates.

Example 1.2
The flotillins (the FLOT-1 and FLOT-2 genes) have been observed to regulate the proto-oncogene ERBB2 *in vitro* (Pust *et al.*, 2013). One may wish to corroborate this *in vivo*. To this end we use gene expression data of a breast cancer study, available as a Bioconductor package: `breastCancerVDX`. From this study the expression levels of probes interrogating the FLOT-1 and ERBB2 genes are retrieved. For clarity of the illustration the FLOT-2 gene is ignored. After centering, the expression levels of the first ERBB2 probe are regressed on those of the four FLOT-1 probes. The R-code below carries out the data retrieval and analysis.

Listing 1.1 R code

```
# load packages
library(Biobase)
library(breastCancerVDX)

# ids of genes FLOT1
idFLOT1 <- which(fData(vdx)[,5] == 10211)

# ids of ERBB2
idERBB2 <- which(fData(vdx)[,5] == 2064)

# get expression levels of probes mapping to FLOT genes
X <- t(exprs(vdx)[idFLOT1,])
```

```
X <- sweep(X, 2, colMeans(X))

# get expression levels of probes mapping to FLOT genes
Y <- t(exprs(vdx)[idERBB2,])
Y <- sweep(Y, 2, colMeans(Y))

# regression analysis
summary(lm(formula = Y[,1] ~ X[,1] + X[,2] + X[,3] + X[,4]))

# correlation among the covariates
cor(X)
```

Prior to the regression analysis, we first assess whether there is collinearity among the FLOT-1 probes through evaluation of the correlation matrix. This reveals a strong correlation ($\hat{\rho} = 0.91$) between the second and third probe. All other cross-correlations do not exceed the 0.20 (in an absolute sense). Hence, there is collinearity among the columns of the design matrix in the to-be-performed regression analysis.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.0000     0.0633  0.0000   1.0000
X[, 1]        0.1641     0.0616  2.6637   0.0081 **
X[, 2]        0.3203     0.3773  0.8490   0.3965
X[, 3]        0.0393     0.2974  0.1321   0.8949
X[, 4]        0.1117     0.0773  1.4444   0.1496
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.175 on 339 degrees of freedom
Multiple R-squared:  0.04834,Adjusted R-squared:  0.03711
F-statistic: 4.305 on 4 and 339 DF,  p-value: 0.002072
```

The output of the regression analysis above shows the first probe to be significantly associated to the expression levels of ERBB2. The collinearity of the second and third probe reveals itself in the standard errors of the effect size: for these probes the standard error is much larger than those of the other two probes. This reflects the uncertainty in the estimates. Regression analysis has difficulty to decide to which covariate the explained proportion of variation in the response should be attributed. The large standard error of these effect sizes propagates to the testing as the Wald test statistic is the ratio of the estimated effect size and its standard error. Collinear covariates are thus less likely to pass the significance threshold.   □

The case of two (or multiple) covariates being perfectly linearly dependent is referred as *super-collinearity*. The rank of a high-dimensional design matrix is maximally equal to n: $\text{rank}(\mathbf{X}) \leq n$. Consequently, the dimension of subspace spanned by the columns of $\mathbf{X}$ is smaller than or equal to n. As $p > n$, this implies that columns of $\mathbf{X}$ are linearly dependent. Put differently, a high-dimensional $\mathbf{X}$ suffers from super-collinearity.

Example 1.3 *Super-collinearity*
Consider the design matrix:

$$\mathbf{X} \;\; = \;\; \begin{pmatrix} 1 & -1 & 2 \\ 1 & 0 & 1 \\ 1 & 2 & -1 \\ 1 & 1 & 0 \end{pmatrix}$$

The columns of $\mathbf{X}$ are linearly dependent: the first column is the row-wise sum of the other two columns. The rank (more correct, the column rank) of a matrix is the dimension of space spanned by the column vectors. Hence, the rank of $\mathbf{X}$ is equal to the number of linearly independent columns: $\text{rank}(\mathbf{X}) = 2$. □

Super-collinearity of an $(n \times p)$-dimensional design matrix $\mathbf{X}$ implies* that the rank of the $(p \times p)$-dimensional matrix $\mathbf{X}^\top \mathbf{X}$ is smaller than p, and, consequently, it is singular. A square matrix that does

---

*If the (column) rank of $\mathbf{X}$ is smaller than p, there exists a non-trivial $\mathbf{v} \in \mathbb{R}^p$ such that $\mathbf{X}\mathbf{v} = \mathbf{0}_p$. Multiplication of this inequality by $\mathbf{X}^\top$ yields $\mathbf{X}^\top \mathbf{X}\mathbf{v} = \mathbf{0}_p$. As $\mathbf{v} \neq \mathbf{0}_p$, this implies that $\mathbf{X}^\top \mathbf{X}$ is not invertible.

not have an inverse is called *singular*. A matrix $\mathbf{A}$ is singular if and only if its determinant is zero: $\det(\mathbf{A}) = 0$.

Example 1.4 *Singularity*
Consider the matrix $\mathbf{A}$ given by:

$$\mathbf{A} \;=\; \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}$$

Clearly, $\det(\mathbf{A}) = a_{11}a_{22} - a_{12}a_{21} = 1 \times 4 - 2 \times 2 = 0$. Hence, $\mathbf{A}$ is singular and its inverse is undefined.□

As $\det(\mathbf{A})$ is equal to the product of the eigenvalues ν_j of $\mathbf{A}$, the matrix $\mathbf{A}$ is singular if one (or more) of the eigenvalues of $\mathbf{A}$ is zero. To see this, consider the spectral decomposition of $\mathbf{A}$:

$$\mathbf{A} \;=\; \sum_{j=1}^{p} \nu_j \, \mathbf{v}_j \, \mathbf{v}_j^{\top},$$

where $\mathbf{v}_j$ is the eigenvector corresponding to ν_j. The inverse of $\mathbf{A}$ is then:

$$\mathbf{A}^{-1} \;=\; \sum_{j=1}^{p} \nu_j^{-1} \, \mathbf{v}_j \, \mathbf{v}_j^{\top}.$$

The right-hand side is undefined if $\nu_j = 0$ for any j.

Example 1.4 *Singularity (continued)*
Revisit Example 1.4. Matrix $\mathbf{A}$ has eigenvalues $\nu_1 = 5$ and $\nu_2 = 0$. According to the spectral decomposition, the inverse of $\mathbf{A}$ is:

$$\mathbf{A}^{-1} \;=\; \frac{1}{5} \mathbf{v}_1 \, \mathbf{v}_1^{\top} + \frac{1}{0} \mathbf{v}_2 \, \mathbf{v}_2^{\top}.$$

This expression is undefined as we divide by zero in the second summand on the right-hand side.   □

In summary, the columns of a high-dimensional design matrix $\mathbf{X}$ are linearly dependent and this super-collinearity causes $\mathbf{X}^{\top}\mathbf{X}$ to be singular. Now recall the ML estimator of the parameter of the linear regression model:

$$\hat{\boldsymbol{\beta}} \;=\; (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{Y}. \tag{1.4}$$

This estimator is only well-defined if $(\mathbf{X}^{\top}\mathbf{X})^{-1}$ exits. Hence, when $\mathbf{X}$ is high-dimensional the regression parameter $\boldsymbol{\beta}$ cannot be estimated.

Above only the practical consequence of high-dimensionality is presented: the expression $(\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{Y}$ cannot be evaluated numerically. But the problem arising from the high-dimensionality of the data is more fundamental. To appreciate this, consider the normal equations:

$$\mathbf{X}^{\top}\mathbf{X}\boldsymbol{\beta} \;=\; \mathbf{X}^{\top}\mathbf{Y}.$$

The matrix $\mathbf{X}^{\top}\mathbf{X}$ is of rank n, while $\boldsymbol{\beta}$ is a vector of length p. Hence, while there are p unknowns, the system of linear equations from which these are to be solved effectively comprises n degrees of freedom. If $p > n$, the vector $\boldsymbol{\beta}$ cannot uniquely be determined from this system of equations. To make this more specific let U be the n-dimensional space spanned by the columns of $\mathbf{X}$ and the $p - n$-dimensional space V be orthogonal complement of U, i.e. $V = U^{\perp}$. Then, $\mathbf{X}\mathbf{v} = \mathbf{0}_p$ for all $\mathbf{v} \in V$. So, V is the non-trivial null space of $\mathbf{X}$. Consequently, as $\mathbf{X}^{\top}\mathbf{X}\mathbf{v} = \mathbf{X}^{\top}\mathbf{0}_p = \mathbf{0}_n$, the solution of the normal equations is:

$$\hat{\boldsymbol{\beta}} \;=\; (\mathbf{X}^{\top}\mathbf{X})^{-}\mathbf{X}^{\top}\mathbf{Y} + \mathbf{v} \qquad \text{for all } \mathbf{v} \in V,$$

where $\mathbf{A}^{-}$ denotes the Moore-Penrose inverse of the matrix $\mathbf{A}$, which is defined as:

$$\mathbf{A}^{-} \;=\; \sum_{j=1}^{p} \nu_j^{-1} I_{\{\nu_j \neq 0\}} \, \mathbf{v}_j \, \mathbf{v}_j^{\top}.$$

The solution of the normal equations is thus only determined up to an element from a non-trivial space V, and there is no unique estimator of the regression parameter.

To obtain an estimate of the regression parameter $\boldsymbol{\beta}$ when $\mathbf{X}$ is (close to) super-collinearity, Hoerl and Kennard (1970) proposed an ad-hoc fix to resolve the (almost) singularity of $\mathbf{X}^\top \mathbf{X}$. Simply replace $\mathbf{X}^\top \mathbf{X}$ by $\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}$ with $\lambda \in [0, \infty)$. The scalar λ is a tuning parameter, henceforth called the *penalty parameter*.

Example 1.3 *Super-collinearity (continued)*
Recall the super-collinear design matrix $\mathbf{X}$ of Example 1.3. Then, for (say) $\lambda = 1$:

$$\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp} \quad = \quad \begin{pmatrix} 5 & 2 & 2 \\ 2 & 7 & -4 \\ 2 & -4 & 7 \end{pmatrix}.$$

The eigenvalues of this matrix are 11, 7, and 1. Hence, $\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}$ has no zero eigenvalue and its inverse is well-defined. □

With the ad-hoc fix for the singularity of $\mathbf{X}^\top \mathbf{X}$, Hoerl and Kennard (1970) proceed to define the *ridge regression estimator*:

$$\hat{\boldsymbol{\beta}}(\lambda) \quad = \quad (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y}, \tag{1.5}$$

for $\lambda \in [0, \infty)$. Clearly, this is – for λ strictly positive – a well-defined estimator, even if $\mathbf{X}$ is high-dimensional. However, each choice of λ leads to a different ridge regression estimate. The set of all ridge regression estimates $\{\hat{\boldsymbol{\beta}}(\lambda) : \lambda \in [0, \infty)\}$ is called the *solution* or *regularization path* of the ridge estimator.

Example 1.3 *Super-collinearity (continued)*
Recall the super-collinear design matrix $\mathbf{X}$ of Example 1.3. Suppose that the corresponding response vector is $\mathbf{Y} = (1.3, -0.5, 2.6, 0.9)^\top$. The ridge regression estimates for, e.g. $\lambda = 1, 2$, and 10 are then:

$$\begin{aligned} \hat{\boldsymbol{\beta}}(1) &= (0.614, 0.548, 0.066)^\top, \\ \hat{\boldsymbol{\beta}}(2) &= (0.537, 0.490, 0.048)^\top, \\ \hat{\boldsymbol{\beta}}(10) &= (0.269, 0.267, 0.002)^\top. \end{aligned}$$

The full solution path of the ridge estimator is plotted in Figure 1.1.

Having obtained an estimate of the regression parameter $\boldsymbol{\beta}$, one can define the fit $\widehat{\mathbf{Y}}$. It is defined analogous to the standard case:

$$\widehat{\mathbf{Y}}(\lambda) \quad = \quad \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda) \quad = \quad \mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y} \quad := \quad \mathbf{H}(\lambda)\mathbf{Y}.$$

Previously, when using the ML estimator, the fit could be understood as a projection of $\mathbf{Y}$ onto the subspace spanned by the columns of $\mathbf{X}$. The fit $\widehat{\mathbf{Y}}(\lambda)$ corresponding to the ridge estimator is not a projection of $\mathbf{Y}$ onto $\mathbf{X}$ (confer Exercise 1.3 a). Consequently, the 'ridge residuals' $\mathbf{Y} - \widehat{\mathbf{Y}}(\lambda)$ are not orthogonal to the fit $\widehat{\mathbf{Y}}(\lambda)$ (confer Exercise 1.3 b).

1.3 Eigenvalue shrinkage

The effect of the ridge penalty may also studied from the perspective of singular values. Let the singular value decomposition of the $(n \times p)$-dimensional design matrix $\mathbf{X}$ be:

$$\mathbf{X} \quad = \quad \mathbf{U}_x \mathbf{D}_x \mathbf{V}_x^\top,$$

where $\mathbf{D}_x$ an $(n \times n)$-dimensional diagonal matrix with the singular values, $\mathbf{U}_x$ an $(n \times n)$-dimensional matrix with columns containing the left singular vectors (denoted $\mathbf{u}_i$), and $\mathbf{V}_x$ a $(p \times n)$-dimensional matrix with columns containing the right singular vectors (denoted $\mathbf{v}_i$). The columns of $\mathbf{U}_x$ and $\mathbf{V}_x$ are orthogonal: $\mathbf{U}_x^\top \mathbf{U}_x = \mathbf{I}_{nn} = \mathbf{V}_x^\top \mathbf{V}_x$.

Figure 1.1: Solution path of the ridge estimator and its variance. The left panel shows the solution path of the ridge estimator for the data of Example 1.3. In the right panel the corresponding variance of the ridge estimator is plotted against the (logarithm of the) penalty parameter.

The OLS estimator can then be rewritten in terms of the SVD-matrices as:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} \\
&= (\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{U}_x\mathbf{D}_x\mathbf{V}_x^\top)^{-1}\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y} \\
&= (\mathbf{V}_x\mathbf{D}_x^2\mathbf{V}_x^\top)^{-1}\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y} \\
&= \mathbf{V}_x\mathbf{D}_x^{-2}\mathbf{V}_x^\top\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y} \\
&= \mathbf{V}_x\mathbf{D}_x^{-2}\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y},
\end{aligned}
$$

where $\mathbf{D}_x^{-2}\mathbf{D}_x$ is not simplified further to emphasize the effect of the ridge penalty. Similarly, the ridge estimator can be rewritten in terms of the SVD-matrices as:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{Y} \\
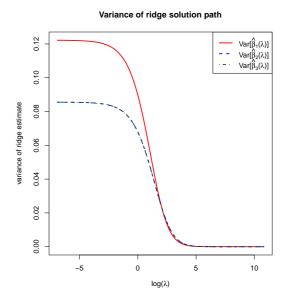&= (\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{U}_x\mathbf{D}_x\mathbf{V}_x^\top + \lambda\mathbf{I}_{pp})^{-1}\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y} \\
&= (\mathbf{V}_x\mathbf{D}_x^2\mathbf{V}_x^\top + \lambda\mathbf{V}_x\mathbf{V}_x^\top)^{-1}\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y} \\
&= \mathbf{V}_x(\mathbf{D}_x^2 + \lambda\mathbf{I}_{nn})^{-1}\mathbf{V}_x^\top\mathbf{V}_x\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y} \\
&= \mathbf{V}_x(\mathbf{D}_x^2 + \lambda\mathbf{I}_{nn})^{-1}\mathbf{D}_x\mathbf{U}_x^\top\mathbf{Y}.
\end{aligned}
$$

Combining the two results and writing $(\mathbf{D}_x)_{jj} = d_{x,jj}$ we have:

$$
d_{x,jj}^{-1} \geq \frac{d_{x,jj}}{d_{x,jj}^2 + \lambda} \qquad \text{for all } \lambda > 0.
$$

Thus, the ridge penalty shrinks the singular values.

Return to the problem of the super-collinearity of $\mathbf{X}$ in the high-dimensional setting ($p > n$). The super-collinearity implies the singularity of $\mathbf{X}^\top\mathbf{X}$ and prevents the calculation of the OLS estimator of the regression coefficients. However, $\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}$ is non-singular, with inverse:

$$
(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1} = \sum_{j=1}^{p}(d_{x,jj}^2 + \lambda)^{-1}\mathbf{v}_j\mathbf{v}_j^\top.
$$

The right-hand side is well-defined for $\lambda > 0$.

1.3.1 Principal components regression

Principal component regression is a close relative to ridge regression that can also be applied in a high-dimensional context. Principal components regression explains the response not by the covariates themselves but by linear combinations of the covariates as defined by the principal components of $\mathbf{X}$. Let $\mathbf{UDV}^\top$ be the singular value decomposition of $\mathbf{X}$. The i-th principal component of $\mathbf{X}$ is then $\mathbf{Xv}_i$, henceforth denoted $\mathbf{z}_i$. Let $\mathbf{Z}_k$ be the matrix of the first k principal components, i.e. $\mathbf{Z}_k = \mathbf{XV}_k$ where $\mathbf{V}_k$ contains the first k right singular vectors as columns. Principal components regression then amounts to regressing the response $\mathbf{Y}$ onto $\mathbf{Z}_k$, that is, it fits the model $\mathbf{Y} = \mathbf{Z}_k\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$. The least squares estimator of $\boldsymbol{\gamma}$ then is (with some abuse of notation):

$$
\begin{aligned}
\hat{\boldsymbol{\gamma}} &= (\mathbf{Z}_k^\top \mathbf{Z}_k)^{-1}\mathbf{Z}_k^\top \mathbf{Y} \;=\; (\mathbf{V}_k^\top \mathbf{X}^\top \mathbf{X}\mathbf{V}_k)^{-1}\mathbf{V}_k^\top \mathbf{X}^\top \mathbf{Y} \\
&= (\mathbf{V}_k^\top \mathbf{V}\mathbf{D}\mathbf{U}^\top \mathbf{U}\mathbf{D}\mathbf{V}^\top \mathbf{V}_k)^{-1}\mathbf{V}_k^\top \mathbf{V}\mathbf{D}\mathbf{U}^\top \mathbf{Y} \\
&= (\mathbf{I}_{kn}\mathbf{D}^2\mathbf{I}_{nk})^{-1}\mathbf{I}_{kn}\mathbf{D}\mathbf{U}^\top \mathbf{Y} \\
&= \mathbf{D}_k^{-2}\widetilde{\mathbf{D}}_k\mathbf{U}^\top \mathbf{Y} \;=\; \widetilde{\mathbf{D}}_k^{-1}\mathbf{U}^\top \mathbf{Y},
\end{aligned}
$$

where $\mathbf{D}_k$ and $\widetilde{\mathbf{D}}_k$ are submatrices of $\mathbf{D}$. The matrix $\mathbf{D}_k$ is obtained from $\mathbf{D}$ by removal of the last $n-p$ rows and columsn, while for $\widetilde{\mathbf{D}}_k$ only the last $n-k$ rows are dropped. Similarly, $\mathbf{I}_{kn}$ and $\mathbf{I}_{nk}$ are obtained from $\mathbf{I}_{nn}$ by removal of the last $n-k$ rows and columns, respectively. The principal component regression estimator of $\boldsymbol{\beta}$ then is $\hat{\boldsymbol{\beta}}_{\mathrm{pcr}} = \mathbf{V}_k\widetilde{\mathbf{D}}_k^{-1}\mathbf{U}^\top \mathbf{Y}$. When k is set equal to the column rank of $\mathbf{X}$, and thus to the rank of $\mathbf{X}^\top \mathbf{X}$, the principal component regression estimator $\hat{\boldsymbol{\beta}}_{\mathrm{pcr}} = (\mathbf{X}^\top \mathbf{X})^-\mathbf{X}^\top \mathbf{Y}$, where $\mathbf{A}^-$ denotes the Moore-Penrose inverse of matrix $\mathbf{A}$.

The relation between ridge and principal component regression becomes clear when their corresponding estimators are written in terms of the singular value decomposition of $\mathbf{X}$:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}_{\mathrm{pcr}} &= \mathbf{V}_x(\mathbf{I}_{nk}\mathbf{D}_x\mathbf{I}_{kn})^{-1}\mathbf{U}_x^\top \mathbf{Y}, \\
\hat{\boldsymbol{\beta}}(\lambda) &= \mathbf{V}_x(\mathbf{D}_x^2 + \lambda\mathbf{I}_{nn})^{-1}\mathbf{D}_x\mathbf{U}_x^\top \mathbf{Y}.
\end{aligned}
$$

Both operate on the singular values of the design matrix. But where principal component regression thresholds the singular values of $\mathbf{X}$, ridge regression shrinks them (depending on their size). Hence, one applies a discrete map on the singular values while the other a continuous one.

1.4 Moments

The first two moments of the ridge regression estimator are derived. Next the performance of the ridge regression estimator is studied in terms of the mean squared error, which combines the first two moments.

1.4.1 Expectation

The left panel of Figure 1.1 shows ridge estimates of the regression parameters converging to zero as the penalty parameter tends to infinity. This behaviour of the ridge estimator does not depend on the specifics of the data set. To see this study the expectation of the ridge estimator:

$$
\begin{aligned}
\mathbb{E}\big[\hat{\boldsymbol{\beta}}(\lambda)\big] &= \mathbb{E}\big[(\mathbf{X}^\top \mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top \mathbf{Y}\big] \\
&= \mathbb{E}\big[(\mathbf{X}^\top \mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top \mathbf{X})(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y}\big] \\
&= \mathbb{E}\big[(\mathbf{X}^\top \mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top \mathbf{X})\,\hat{\boldsymbol{\beta}}\big] \\
&= (\mathbf{X}^\top \mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top \mathbf{X})\,\mathbb{E}(\hat{\boldsymbol{\beta}}) \\
&= (\mathbf{X}^\top \mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top \mathbf{X})\,\boldsymbol{\beta}.
\end{aligned}
$$

Clearly, $\mathbb{E}\big[\hat{\boldsymbol{\beta}}(\lambda)\big] \neq \boldsymbol{\beta}$ for any $\lambda > 0$. Hence, the ridge estimator is biased.

From the expression above it is clear that the expectation of the ridge estimator vanishes as λ tends to infinity:

$$
\lim_{\lambda\to\infty}\mathbb{E}\big[\hat{\boldsymbol{\beta}}(\lambda)\big] \;=\; \lim_{\lambda\to\infty}(\mathbf{X}^\top \mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top \mathbf{X})\,\boldsymbol{\beta} \;=\; \mathbf{0}_p.
$$

Hence, all regression coefficients are shrunken towards zero as the penalty parameter increases. This also holds for $\mathbf{X}$ with $p > n$. Furthermore, this behaviour is not strictly monotone in λ: $\lambda_a > \lambda_b$ does not necessarily imply $|\hat{\beta}_j(\lambda_a)| < |\hat{\beta}_j(\lambda_b)|$. Upon close inspection this can be witnessed from the ridge solution path of β_3 in Figure 1.1.

Example 1.5 *Orthonormal design matrix*
Consider an orthonormal design matrix $\mathbf{X}$, i.e.:

$$\mathbf{X}^\top \mathbf{X} \;\; = \;\; \mathbf{I}_{pp} \;\; = \;\; (\mathbf{X}^\top \mathbf{X})^{-1}.$$

An example of an orthonormal design matrix would be:

$$\mathbf{X} \;\; = \;\; \frac{1}{2} \begin{pmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

This design matrix is orthonormal as $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_{22}$, which is easily verified:

$$\mathbf{X}^\top \mathbf{X} \;\; = \;\; \frac{1}{4} \begin{pmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix} \;\; = \;\; \frac{1}{4} \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} \;\; = \;\; \mathbf{I}_{22}.$$

In case of an orthonormal design matrix the relation between the OLS and ridge estimator is:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}(\lambda) \;\; &= \;\; (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}^\top \mathbf{Y} \;\; = \;\; (\mathbf{I}_{pp} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}^\top \mathbf{Y} \\
&= \;\; (1+\lambda)^{-1}\mathbf{I}_{pp}\mathbf{X}^\top \mathbf{Y} \qquad = \;\; (1+\lambda)^{-1}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y} \\
&= \;\; (1+\lambda)^{-1}\hat{\boldsymbol{\beta}}.
\end{aligned}$$

Hence, the ridge estimator scales the OLS estimator by a factor. When taking the expectation on both sides, it is evident that the ridge estimator converges to zero as $\lambda \to \infty$. $\qquad\square$

1.4.2 Variance

As for the ML estimate of the regression parameter $\boldsymbol{\beta}$ of model (1.2), we derive the second moment of the ridge estimator. Hereto define:

$$\mathbf{W}_\lambda \;\; = \;\; (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}^\top \mathbf{X}.$$

Using $\mathbf{W}_\lambda$ the ridge estimator $\hat{\boldsymbol{\beta}}(\lambda)$ can be expressed as $\mathbf{W}_\lambda \hat{\boldsymbol{\beta}}$ for:

$$\begin{aligned}
\mathbf{W}_\lambda \hat{\boldsymbol{\beta}} \;\; &= \;\; \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y} \\
&= \;\; \{(\mathbf{X}^\top \mathbf{X})^{-1}[\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]\}^{-1}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y} \\
&= \;\; [\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1}\mathbf{X}^\top \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y} \\
&= \;\; [\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1}\mathbf{X}^\top \mathbf{Y} \\
&= \;\; \hat{\boldsymbol{\beta}}(\lambda).
\end{aligned}$$

The linear operator $\mathbf{W}_\lambda$ thus transforms the ML estimator of the regression parameter into the ridge estimator.

It is now easily seen that:

$$\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{\beta}}(\lambda)] \;\; &= \;\; \mathrm{Var}[\mathbf{W}_\lambda \hat{\boldsymbol{\beta}}] \qquad\qquad = \;\; \mathbf{W}_\lambda \mathrm{Var}[\hat{\boldsymbol{\beta}}]\mathbf{W}_\lambda^\top \\
&= \;\; \sigma^2 \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{W}_\lambda^\top \;\; = \;\; \sigma^2 [\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1}\mathbf{X}^\top \mathbf{X}\{[\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1}\}^\top,
\end{aligned}$$

in which we have used $\mathrm{Var}(\mathbf{A}\mathbf{Y}) = \mathbf{A}\mathrm{Var}(\mathbf{Y})\mathbf{A}^\top$ for a non-random matrix $\mathbf{A}$, the fact that $\mathbf{W}_\lambda$ is non-random, and $\mathrm{Var}[\hat{\boldsymbol{\beta}}] = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}$.

Like the expectation the variance of the ridge estimator vanishes as λ tends to infinity:

$$\lim_{\lambda \to \infty} \text{Var}\big[\hat{\boldsymbol{\beta}}(\lambda)\big] \quad = \quad \lim_{\lambda \to \infty} \sigma^2 \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{W}_\lambda^\top \quad = \quad \mathbf{0}_{pp}.$$

Hence, the variance of the ridge regression coefficient estimates decreases towards zero as the penalty parameter becomes large. This is illustrated in the right panel of Figure 1.1 for the data of Example 1.3.
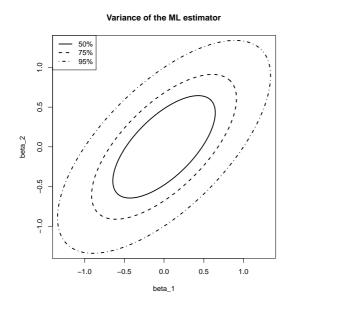
With an explicit expression of the variance of the ridge estimator at hand, we can compare it to that of the OLS estimator:

$$
\begin{aligned}
\text{Var}[\hat{\boldsymbol{\beta}}] - \text{Var}[\hat{\boldsymbol{\beta}}(\lambda)] &= \sigma^2 [(\mathbf{X}^\top \mathbf{X})^{-1} - \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{W}_\lambda^\top] \\
&= \sigma^2 \mathbf{W}_\lambda \{[\mathbf{I} + \lambda (\mathbf{X}^\top \mathbf{X})^{-1}](\mathbf{X}^\top \mathbf{X})^{-1}[\mathbf{I} + \lambda (\mathbf{X}^\top \mathbf{X})^{-1}]^\top - (\mathbf{X}^\top \mathbf{X})^{-1}\} \mathbf{W}_\lambda^\top \\
&= \sigma^2 \mathbf{W}_\lambda [2\,\lambda\,(\mathbf{X}^\top \mathbf{X})^{-2} + \lambda^2 (\mathbf{X}^\top \mathbf{X})^{-3}] \mathbf{W}_\lambda^\top \\
&= \sigma^2 [\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1} [2\,\lambda\,\mathbf{I}_{pp} + \lambda^2 (\mathbf{X}^\top \mathbf{X})^{-1}] \{[\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1}\}^\top.
\end{aligned}
$$

The difference is non-negative definite as each component in the matrix product is non-negative definite. Hence, the variance of the ML estimator exceeds (in the positive definite ordering) that of the ridge estimator:

$$\text{Var}[\hat{\boldsymbol{\beta}}] \quad \succeq \quad \text{Var}[\hat{\boldsymbol{\beta}}(\lambda)], \tag{1.6}$$

with the inequality being strict if $\lambda > 0$. In other words, the variance of the ML estimator is larger than that of the ridge estimator (in the sense that their difference is non-negative definite). The variance inequality (1.6) can be interpreted in terms of the stochastic behaviour of the estimator. This is illustrated by the next example.

Figure 1.2: Level sets of the distribution of the ML (left panel) and ridge (right panel) regression estimators.

Example 1.6 *Variance comparison*
Consider the design matrix:

$$\mathbf{X} \quad = \quad \begin{pmatrix} -1 & 2 \\ 0 & 1 \\ 2 & -1 \\ 1 & 0 \end{pmatrix}.$$

The variances of the ML and ridge (with $\lambda = 1$) estimates of the regression coefficients then are:

$$\text{Var}(\hat{\boldsymbol{\beta}}) \quad = \quad \sigma^2 \begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{pmatrix} \qquad \text{and} \qquad \text{Var}[\hat{\boldsymbol{\beta}}(\lambda)] \quad = \quad \sigma^2 \begin{pmatrix} 0.1524 & 0.0698 \\ 0.0698 & 0.1524 \end{pmatrix}.$$

These variances can be used to construct levels sets of the distribution of the estimates. The level sets that contain 50%, 75% and 95% of the distribution of the ML and ridge estimates are plotted in Figure 1.2. In line with inequality (1.6) the level sets of the ridge estimate are smaller than that of the ML estimate: it thus varies less. $\square$

Example 1.5 *Orthonormal design matrix (continued)*
Assume the design matrix $\mathbf{X}$ is orthonormal. Then, $\mathrm{Var}[\hat{\boldsymbol{\beta}}] = \sigma^2 \mathbf{I}_{pp}$ and

$$\mathrm{Var}[\hat{\boldsymbol{\beta}}(\lambda)] \quad = \quad \sigma^2 \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{W}_\lambda^\top \quad = \quad \sigma^2 [\mathbf{I}_{pp} + \lambda \mathbf{I}_{pp}]^{-1} \mathbf{I}_{pp} \{[\mathbf{I}_{pp} + \lambda \mathbf{I}_{pp}]^{-1}\}^\top \quad = \quad \sigma^2 (1+\lambda)^{-2} \mathbf{I}_{pp}.$$

As the penalty parameter λ is non-negative the former exceeds the latter. In particular, this expression vanishes as $\lambda \to \infty$. $\square$

The full distribution of the ridge regression estimator is now known. The estimator, $\hat{\boldsymbol{\beta}}(\lambda) = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y}$ is a linear estimator, linear in $\mathbf{Y}$. As $\mathbf{Y}$ is normally distributed, so is $\hat{\boldsymbol{\beta}}(\lambda)$. Moreover, the normal distribution is fully characterized by its first two moments, which are available. Hence:

$$\hat{\boldsymbol{\beta}}(\lambda) \quad \sim \quad \mathcal{N}\big((\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{X} \boldsymbol{\beta}, \; \sigma^2 [\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1} \mathbf{X}^\top \mathbf{X} \{[\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1}\}^\top \big).$$

Given λ and $\boldsymbol{\beta}$, the random behavior of the estimator is thus known.

1.4.3   Mean squared error

Previously, we motivated the ridge estimator as an ad hoc solution to collinearity. An alternative motivation comes from studying the Mean Squared Error (MSE) of the ridge regression estimator: for a suitable choice of λ the ridge regression estimator may outperform the ML regression estimator in terms of the MSE. Before we prove this, we first derive the MSE of the ridge estimator and quote some auxiliary results.

Recall that (in general) for any estimator of a parameter θ:

$$\mathrm{MSE}(\hat{\theta}) \quad = \quad \mathbb{E}[(\hat{\theta} - \theta)^2] \quad = \quad \mathrm{Var}(\hat{\theta}) + [\mathrm{Bias}(\hat{\theta})]^2.$$

Hence, the MSE is a measure of the quality of the estimator.

The MSE of the ridge estimator is:

$$
\begin{aligned}
\mathrm{MSE}[\hat{\boldsymbol{\beta}}(\lambda)] \quad &= \quad \mathbb{E}[(\mathbf{W}_\lambda \hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top (\mathbf{W}_\lambda \hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] \\
&= \quad \mathbb{E}(\hat{\boldsymbol{\beta}}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \hat{\boldsymbol{\beta}}) - \mathbb{E}(\boldsymbol{\beta}^\top \mathbf{W}_\lambda \hat{\boldsymbol{\beta}}) - \mathbb{E}(\hat{\boldsymbol{\beta}}^\top \mathbf{W}_\lambda^\top \boldsymbol{\beta}) + \mathbb{E}(\boldsymbol{\beta}^\top \boldsymbol{\beta}) \\
&= \quad \mathbb{E}(\hat{\boldsymbol{\beta}}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \hat{\boldsymbol{\beta}}) - \mathbb{E}(\boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \hat{\boldsymbol{\beta}}) - \mathbb{E}(\hat{\boldsymbol{\beta}}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \boldsymbol{\beta}) + \mathbb{E}(\boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \boldsymbol{\beta}) \\
&\quad\quad - \mathbb{E}(\boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \boldsymbol{\beta}) + \mathbb{E}(\boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \hat{\boldsymbol{\beta}}) + \mathbb{E}(\hat{\boldsymbol{\beta}}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \boldsymbol{\beta}) \\
&\quad\quad - \mathbb{E}(\boldsymbol{\beta}^\top \mathbf{W}_\lambda \hat{\boldsymbol{\beta}}) - \mathbb{E}(\hat{\boldsymbol{\beta}}^\top \mathbf{W}_\lambda^\top \boldsymbol{\beta}) + \mathbb{E}(\boldsymbol{\beta}^\top \boldsymbol{\beta}) \\
&= \quad \mathbb{E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] \\
&\quad\quad - \boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \boldsymbol{\beta} + \boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \boldsymbol{\beta} + \boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda \boldsymbol{\beta} \\
&\quad\quad - \boldsymbol{\beta}^\top \mathbf{W}_\lambda \boldsymbol{\beta} - \boldsymbol{\beta}^\top \mathbf{W}_\lambda^\top \boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{\beta} \\
&= \quad \mathbb{E}\big\{ (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top \mathbf{W}_\lambda^\top \mathbf{W}_\lambda (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \big\} + \boldsymbol{\beta}^\top (\mathbf{W}_\lambda - \mathbf{I}_{pp})^\top (\mathbf{W}_\lambda - \mathbf{I}_{pp}) \boldsymbol{\beta} \\
&= \quad \sigma^2 \, \mathrm{tr}\big\{ \mathbf{W}_\lambda (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{W}_\lambda^\top \big\} + \boldsymbol{\beta}^\top (\mathbf{W}_\lambda - \mathbf{I}_{pp})^\top (\mathbf{W}_\lambda - \mathbf{I}_{pp}) \boldsymbol{\beta}. \quad\quad (1.7)
\end{aligned}
$$

In the last step we have used $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2 [\mathbf{X}^\top \mathbf{X}]^{-1})$ and the expectation of the quadratic form of a multivariate random variable $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\mu}_\varepsilon, \boldsymbol{\Sigma}_\varepsilon)$ for some nonrandom symmetric positive definite matrix $\boldsymbol{\Lambda}$ is (cf. Mathai and Provost 1992):

$$\mathbb{E}(\boldsymbol{\varepsilon}^\top \boldsymbol{\Lambda} \boldsymbol{\varepsilon}) \quad = \quad \mathrm{tr}(\boldsymbol{\Lambda} \boldsymbol{\Sigma}_\varepsilon) + \boldsymbol{\mu}_\varepsilon^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_\varepsilon,$$

of course replacing $\boldsymbol{\varepsilon}$ by $\hat{\boldsymbol{\beta}}$ in this expectation. The first summand in the final derived expression for $\mathrm{MSE}[\hat{\boldsymbol{\beta}}(\lambda)]$ is the sum of the variances of the ridge estimator, while the second summand can be thought of the "squared bias" of the ridge estimator. In particular, $\lim_{\lambda \to \infty} \mathrm{MSE}[\hat{\boldsymbol{\beta}}(\lambda)] = \boldsymbol{\beta}^\top \boldsymbol{\beta}$, which is the squared biased for an estimator that equals zero (as does the ridge estimator in the limit).

Example 1.7 *Orthonormal design matrix*
Assume the design matrix $\mathbf{X}$ is orthonormal. Then, $\mathrm{MSE}[\hat{\boldsymbol{\beta}}] = p\,\sigma^2$ and

$$\mathrm{MSE}[\hat{\boldsymbol{\beta}}(\lambda)] \;\;=\;\; \frac{p\,\sigma^2}{(1+\lambda)^2} + \frac{\lambda^2}{(1+\lambda)^2}\boldsymbol{\beta}^\top\boldsymbol{\beta}.$$

The latter achieves its minimum at: $\lambda = p\sigma^2/\boldsymbol{\beta}^\top\boldsymbol{\beta}$.   □

The following theorem and proposition are required for the proof of the main result.

Theorem 1.1 *(Theorem 1 of Theobald, 1974)*
Let $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ be (different) estimators of $\boldsymbol{\theta}$ with second order moments:

$$\mathbf{M}_k \;\;=\;\; \mathbb{E}[(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta})^\top] \qquad \text{for } k = 1, 2,$$

and

$$\mathrm{MSE}(\hat{\boldsymbol{\theta}}_k) \;\;=\;\; \mathbb{E}[(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta})^\top \mathbf{A}(\hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta})] \qquad \text{for } k = 1, 2,$$

where $\mathbf{A} \succeq 0$. Then, $\mathbf{M}_1 - \mathbf{M}_2 \succeq 0$ if and only if $\mathrm{MSE}(\hat{\boldsymbol{\theta}}_1) - \mathrm{MSE}(\hat{\boldsymbol{\theta}}_2) \geq 0$ for all $\mathbf{A} \succeq 0$.

Proposition 1.1 *(Farebrother, 1976)*
Let $\mathbf{A}$ be a $(p \times p)$-dimensional, positive definite matrix, $\mathbf{b}$ be a nonzero p dimensional vector, and $c \in \mathbb{R}_+$. Then, $c\mathbf{A} - \mathbf{b}\mathbf{b}^\top \succ 0$ if and only if $\mathbf{b}^\top\mathbf{A}^{-1}\mathbf{b} > c$.

We are now ready to proof the main result, formalized as Theorem 1.2, that for some λ the ridge regression estimator yields a lower MSE than the ML regression estimator.

Theorem 1.2 *(Theorem 2 of Theobald, 1974)*
There exists $\lambda > 0$ such that $\mathrm{MSE}[\hat{\boldsymbol{\beta}}(\lambda)] < \mathrm{MSE}[\hat{\boldsymbol{\beta}}(0)] = \mathrm{MSE}[\hat{\boldsymbol{\beta}}]$.

Proof The second order moment matrix of the ridge estimator is:

$$\begin{aligned}
\mathbf{M}(\lambda) &:= \mathbb{E}[(\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta})(\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta})^\top] \\
&= \mathbb{E}\{\hat{\boldsymbol{\beta}}(\lambda)[\hat{\boldsymbol{\beta}}(\lambda)]^\top\} - \mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda)]\{\mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda)]\}^\top + \mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta})]\{\mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta})]\}^\top \\
&= \mathrm{Var}[\hat{\boldsymbol{\beta}}(\lambda)] + \mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta})]\{\mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta})]\}^\top.
\end{aligned}$$

Then:

$$\begin{aligned}
\mathbf{M}(0) - \mathbf{M}(\lambda) &= \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} - \sigma^2\mathbf{W}_\lambda(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{W}_\lambda^\top] \\
&\quad -(\mathbf{W}_\lambda - \mathbf{I}_{pp})\boldsymbol{\beta}\boldsymbol{\beta}^\top(\mathbf{W}_\lambda - \mathbf{I}_{pp})^\top \\
&= \sigma^2\mathbf{W}_\lambda[2\,\lambda\,(\mathbf{X}^\top\mathbf{X})^{-2} + \lambda^2(\mathbf{X}^\top\mathbf{X})^{-3}]\mathbf{W}_\lambda^\top \\
&\quad -\lambda^2[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}\boldsymbol{\beta}\boldsymbol{\beta}^\top\{[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}\}^\top \\
&= \sigma^2[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}[2\,\lambda\,\mathbf{I}_{pp} + \lambda^2(\mathbf{X}^\top\mathbf{X})^{-1}]\{[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}\}^\top \\
&\quad -\lambda^2[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}\boldsymbol{\beta}\boldsymbol{\beta}^\top\{[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}\}^\top \\
&= \lambda[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}[2\,\sigma^2\,\mathbf{I}_{pp} + \lambda\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} - \lambda\boldsymbol{\beta}\boldsymbol{\beta}^\top]\{[\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp}]^{-1}\}^\top.
\end{aligned}$$

This is positive definite if and only if $2\,\sigma^2\,\mathbf{I}_{pp} + \lambda\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1} - \lambda\boldsymbol{\beta}\boldsymbol{\beta}^\top \succ 0$. Hereto it suffices to show that $2\,\sigma^2\,\mathbf{I}_{pp} - \lambda\boldsymbol{\beta}\boldsymbol{\beta}^\top \succ 0$. By Proposition 1.1 this holds for λ such that $2\sigma^2(\boldsymbol{\beta}^\top\boldsymbol{\beta})^{-1} > \lambda$. For these λ, we thus have $\mathbf{M}(0) - \mathbf{M}(\lambda)$. Application of Theorem 1.1 now concludes the proof. ■

This result of Theobald (1974) is generalized by Farebrother (1976) to the class of design matrices $\mathbf{X}$ with $\mathrm{rank}(\mathbf{X}) < p$.

Theorem 1.2 can be used to illustrate that the ridge regression estimator strikes a balance between the bias and variance. This is illustrated in the left panel of Figure 1.3. For small λ, the variance of the ridge estimator dominates the MSE. This may be understood when realizing that in this domain of λ the ridge estimator is close to the unbiased ML regression estimator. For large λ, the variance vanishes
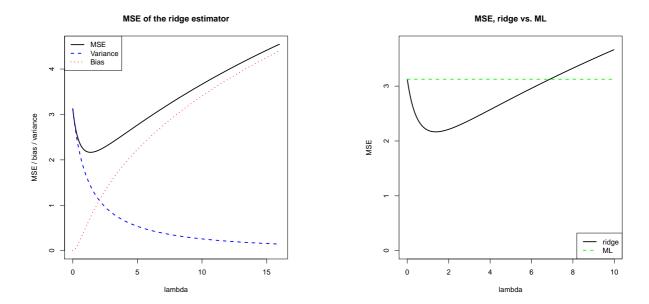
Figure 1.3: Left panel: mean squared error, and its 'bias' and 'variance' parts, of the ridge regression estimator (for artificial data). Right panel: mean squared error of the ridge and ML estimator of the regression coefficient vector (for the same artificial data).

and the bias dominates the MSE. For small enough values of λ, the decrease in variance of the ridge regression estimator exceeds the increase in its bias. As the MSE is the sum of these two, the MSE first decreases as λ moves away from zero. In particular, as $\lambda = 0$ corresponds to the ML regression estimator, the ridge regression estimator yields a lower MSE for these values of λ. In the right panel of Figure 1.3 $\mathrm{MSE}[\hat{\boldsymbol{\beta}}(\lambda)] < \mathrm{MSE}[\hat{\boldsymbol{\beta}}(0)]$ for $\lambda < 7$ (roughly) and the ridge estimator outperforms the ML estimator.

Besides another motivation behind the ridge regression estimator, the use of Theorem 1.2 is limited. The optimal choice of λ depends on the quantities $\boldsymbol{\beta}$ and σ^2. These are unknown in practice. Then, the penalty parameter is chosen in a data-driven fashion by means of cross-validation (see Section 1.9.2).

Remark 1.1
Theorem 1.2 can also be used to conclude on the biasedness of the ridge regression estimator. The Gauss-Markov theorem (Rao, 1973) states (under some assumptions) that the ML regression estimator is the best linear unbiased estimator (BLUE) with the smallest MSE. As the ridge regression estimator is a linear estimator and outperforms (in terms of MSE) this ML estimator, it must be biased (for it would otherwise refute the Gauss-Markov theorem).

1.5 Constrained estimation

The ad-hoc fix of Hoerl and Kennard (1970) to super-collinearity of the design matrix (and, consequently the singularity of the matrix $\mathbf{X}^\top \mathbf{X}$) has been motivated post-hoc. The ridge estimator minimizes the *ridge loss function*, which is defined as:

$$\mathcal{L}_{\mathrm{ridge}}(\boldsymbol{\beta}; \lambda) \quad = \quad \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \quad = \quad \sum_{i=1}^n (Y_i - \mathbf{X}_{i*}\boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p \beta_j^2. \qquad (1.8)$$

This loss function is the traditional sum-of-squares augmented with a *penalty*. The particular form of the penalty, $\lambda\|\boldsymbol{\beta}\|_2^2$ is referred to as the *ridge penalty* and λ as the *penalty parameter*. For $\lambda = 0$, minimization of the ridge loss function yields the ML estimator. For any $\lambda > 0$, the ridge penalty contributes to the loss function, affecting its minimum and its location. The minimum of the sum-of-squares is well-known. The

minimum of the ridge penalty is attained at $\boldsymbol{\beta} = \mathbf{0}_p$ whenever $\lambda > 0$. The $\boldsymbol{\beta}$ that minimizes $\mathcal{L}_{\mathrm{ridge}}(\boldsymbol{\beta}; \lambda)$ then balances the sum-of-squares and the penalty. The effect of the penalty in this balancing act is to shrink the regression coefficients towards zero, its minimum. In particular, the larger λ, the larger the contribution of the penalty to the loss function, the stronger the tendency to shrink non-zero regression coefficients to zero (and decrease the contribution of the penalty to the loss function). This motivates the name 'penalty' as non-zero elements of $\boldsymbol{\beta}$ increase (or penalize) the loss function.

To verify that the ridge estimator indeed minimizes the ridge loss function, proceed as usual. Take the derivative with respect to $\boldsymbol{\beta}$:

$$\frac{\partial}{\partial \boldsymbol{\beta}} \mathcal{L}_{\mathrm{ridge}}(\boldsymbol{\beta}; \lambda) \quad = \quad -2\,\mathbf{X}^{\top}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + 2\,\lambda\,\mathbf{I}_{pp}\,\boldsymbol{\beta} \quad = \quad -2\,\mathbf{X}^{\top}\mathbf{Y} + 2\,(\mathbf{X}^{\top}\mathbf{X} + \lambda\,\mathbf{I}_{pp})\boldsymbol{\beta}.$$

Equate the derivative to zero and solve for $\boldsymbol{\beta}$. This yields the ridge regression estimator.

The ridge estimator is thus a stationary point of the ridge loss function. A stationary point corresponds to a minimum if the Hessian matrix with second order partial derivatives is positive definite. The Hessian of the ridge loss function is

$$\frac{\partial^2}{\partial \boldsymbol{\beta}\,\partial \boldsymbol{\beta}^{\top}} \mathcal{L}_{\mathrm{ridge}}(\boldsymbol{\beta}; \lambda) \quad = \quad 2\,(\mathbf{X}^{\top}\mathbf{X} + \lambda\,\mathbf{I}_{pp}).$$

This Hessian is the sum of the semi-positive definite matrix $\mathbf{X}^{\top}\mathbf{X}$ and the positive definite matrix $\lambda\,\mathbf{I}_{pp}$. Lemma 14.2.4 of Harville (2008) then states that the sum of these matrices is itself a positive definite matrix. Hence, the Hessian is positive definite and the ridge loss function has a stationary point at the ridge estimator, which is a minimum.

The ridge regression estimator minimizes the ridge loss function. It rests to verify that it is a global minimum. To this end we introduce the concept of a convex function. As a prerequisite, a set $\mathcal{S} \subset \mathbb{R}^p$ is called *convex* if for all $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathcal{S}$ their weighted average $\boldsymbol{\beta}_\theta = (1 - \theta)\boldsymbol{\beta}_1 + \theta\boldsymbol{\beta}_2$ for all $\theta \in [0, 1]$ is itself an element of $\mathcal{S}$, thus $\boldsymbol{\beta}_\theta \in \mathcal{S}$. If for all $\theta \in (0, 1)$, the weighted average $\boldsymbol{\beta}_\theta$ is inside $\mathcal{S}$ and not on its boundary, the set is called *strict convex*. Examples of (strict) convex and nonconvex sets are depicted in Figure 1.4. A function $f(\cdot)$ is *(strict) convex* if the set $\{y : y \geq f(\boldsymbol{\beta})$ for all $\boldsymbol{\beta} \in \mathcal{S}$ for any convex $\mathcal{S}\}$, called the epigraph of $f(\cdot)$, is (strict) convex. Examples of (strict) convex and nonconvex functions are depicted in Figure 1.4. The ridge loss function is the sum of two parabola's: one at least convex and the other a strict convex function in $\boldsymbol{\beta}$. The sum of convex and strict convex function is itself strict convex (confer Lemma 9.4.2 of Fletcher 2008). The ridge loss function is thus strict convex. Theorem 9.4.1 of Fletcher 2008 then warrants, by the strict convexity of the ridge loss function, that the ridge estimator is a global minimum.

From the ridge loss function the limiting behavior of the variance of the ridge regression estimator can be understood. The ridge penalty with its minimum $\boldsymbol{\beta} = \mathbf{0}_p$ does not involve data and, consequently, the variance of its minimum equals zero. With the ridge regression being a compromise between the ML estimator and the minimum of the penalty, so is its variance a compromise of their variances. As λ tends to infinity, the ridge estimator and its variance converge to the minimum and the variance of the minimum, respectively. Hence, in the limit (large λ) the variance of the ridge regression estimator vanishes. Understandably, as the penalty now fully dominates the loss function and, consequently, it does no longer involve data (i.e. randomness).

Above it has been shown that the ridge estimator can be defined as:

$$\hat{\boldsymbol{\beta}}(\lambda) \quad = \quad \arg\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\,\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2. \tag{1.9}$$

This minimization problem can be reformulated into the following constrained optimization problem (illustrated in Figure 1.4):

$$\hat{\boldsymbol{\beta}}(\lambda) \quad = \quad \arg\min_{\|\boldsymbol{\beta}\|_2^2 \leq c} \|\mathbf{Y} - \mathbf{X}\,\boldsymbol{\beta}\|_2^2, \tag{1.10}$$

for some suitable $c > 0$. The constrained optimization problem (1.10) can be solved by means of the Karush-Kuhn-Tucker (KKT) multiplier method, which minimizes a function subject to inequality

Convex sets

Nonconvex sets

Convex functions

Nonconvex functions

Ridge as constrained estimation

Overfitting

Figure 1.4: Top panels show examples of convex (left) and nonconvex (right) sets. Middle panels show examples of convex (left) and nonconvex (right) functions. The left bottom panel illustrates the ridge estimation as a constrained estimation problem. The ellipses represent the contours of the ML loss function, with the blue dot at the center the ML estimate. The circle is the ridge parameter constraint. The red dot is the ridge estimate. It is at the intersection of the ridge constraint and the smallest contour with a non-empty intersection with the constraint. The right bottom panel shows the data corresponding to Example 1.8. The grey line represents the 'true' relationship, while the black line the fitted one.

constraints. The KKT multiplier method states that, under some regularity conditions (all met here), there exists a constant $\nu \geq 0$, called the *multiplier*, such that the solution $\hat{\boldsymbol{\beta}}(\nu)$ of the constrained minimization problem (1.10) satisfies the so-called KKT conditions. The first KKT condition (referred to as the stationarity condition) demands that the gradient (with respect to $\boldsymbol{\beta}$) of the Lagrangian associated with the minimization problem equals zero at the solution $\hat{\boldsymbol{\beta}}(\nu)$. The Lagrangian for problem (1.10) is:

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \nu(\|\boldsymbol{\beta}\|_2^2 - c).$$

The second KKT condition (the complementarity condition) requires that $\nu(\|\hat{\boldsymbol{\beta}}(\nu)\|_2^2 - c) = 0$. If $\nu = \lambda$ and $c = \|\hat{\boldsymbol{\beta}}(\lambda)\|_2^2$, the ridge estimator $\boldsymbol{\beta}(\lambda)$ satisfies both KKT conditions. Hence, both problems have the same solution when $c = \|\hat{\boldsymbol{\beta}}(\lambda)\|_2^2$.

The relevance of viewing the ridge regression estimator as the solution to a constrained estimation problem becomes obvious when considering a typical threat to high-dimensional data analysis: overfitting. *Overfitting* refers to the phenomenon of modelling the noise rather than the signal. In case the true model is parsimonious (few covariates driving the response) and data on many covariates are available, it is likely that a linear combination of all covariates yields a higher likelihood than a combination of the few that are actually related to the response. As only the few covariates related to the response contain the signal, the model involving all covariates then cannot but explain more than the signal alone: it also models the error. Hence, it overfits the data. In high-dimensional settings overfitting is a real threat. The number of explanatory variables exceeds the number of observations. It is thus possible to form a linear combination of the covariates that perfectly explains the response, including the noise.

Large estimates of regression coefficients are often an indication of overfitting. Augmentation of the estimation procedure with a constraint on the regression coefficients is a simple remedy to large parameter estimates. As a consequence it decreases the probability of overfitting. Overfitting is illustrated in the next example.

Example 1.8 *(Overfitting)*
Consider an artificial data set comprising of ten observations on a response Y_i and nine covariates $X_{i,j}$. All covariate data are sampled from the standard normal distribution: $X_{i,j} \sim \mathcal{N}(0,1)$. The response is generated by $Y_i = X_{i,1} + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, 1/4)$. Hence, only the first covariate contributes to the response.

The regression model $Y_i = \sum_{j=1}^9 X_{i,j}\beta_j + \varepsilon_i$ is fitted to the artificial data using R. This yields the regression parameter estimates:

$$\hat{\boldsymbol{\beta}}^\top \quad = \quad (0.048, -2.386, -5.528, 6.243, -4.819, 0.760, -3.345, -4.748, 2.136).$$

As $\boldsymbol{\beta}^\top = (1, 0, \ldots, 0)$, many regression coefficient are clearly over-estimated.

The fitted values $\widehat{Y}_i = \mathbf{X}_i\hat{\boldsymbol{\beta}}$ are plotted against the values of the first covariates in the right bottom panel of Figure 1.4. As a reference the line $x = y$ is added, which represents the 'true' model. The fitted model follows the 'true' relationship. But it also captures the deviations from this line that represent the errors. $\square$

1.6 Bayesian regression

Ridge regression has a close connection to Bayesian linear regression. Bayesian linear regression assumes the parameters $\boldsymbol{\beta}$ and σ^2 to be the random variables, while at the same time considering $\mathbf{X}$ and $\mathbf{Y}$ as fixed. Within the regression context, the conjugate priors of $\boldsymbol{\beta}$ and σ^2 are:

$$\boldsymbol{\beta} \,|\, \sigma^2 \sim \mathcal{N}(\mathbf{0}_p, \sigma^2\lambda^{-1}\mathbf{I}_{pp}) \qquad \text{and} \qquad \sigma^2 \sim \mathcal{IG}(\alpha_0, \beta_0),$$

where $\mathcal{IG}$ denotes the inverse Gamma distribution with shape parameter α_0 and scale parameter β_0. The penalty parameter can be interpreted as the precision of the prior, determining how informative the prior should be. A smaller penalty (i.e. precision) corresponds to a wider prior, and a larger penalty to a more informative, concentrated prior (Figure 1.5).
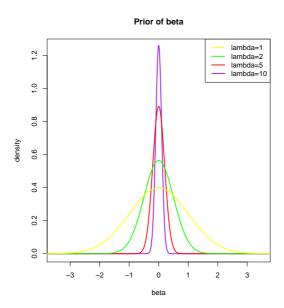
Figure 1.5: Conjugate prior of the regression parameter $\boldsymbol{\beta}$ for various choices of λ, the penalty parameters c.q. precision.

Under the assumption of the conjugate priors above, the joint posterior distribution of $\boldsymbol{\beta}$ and σ^2 is then:

$$
\begin{aligned}
f_{\boldsymbol{\beta},\sigma^2}(\boldsymbol{\beta}, \sigma^2 \,|\, \mathbf{Y}, \mathbf{X}) &= f_Y(\mathbf{Y} \,|\, \mathbf{X}, \boldsymbol{\beta}, \sigma^2)\, f_\beta(\boldsymbol{\beta}|\sigma^2)\, f_\sigma(\sigma^2) \\
&\propto \sigma^{-n} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \right] \\
&\quad \times \sigma^{-p} \exp\left[-\frac{1}{2\sigma^2}\lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} \right] \times [\sigma^2]^{-\alpha_0-1} \exp\left[-\frac{\beta_0}{2\sigma^2} \right].
\end{aligned}
$$

As

$$
\begin{aligned}
(\mathbf{Y} &- \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} \\
&= \mathbf{Y}^\top\mathbf{Y} - \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{Y} - \mathbf{Y}^\top\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} + \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} \\
&= \mathbf{Y}^\top\mathbf{Y} - \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{Y} \\
&\quad - \mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})\boldsymbol{\beta} + \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})\boldsymbol{\beta} \\
&= \mathbf{Y}^\top\mathbf{Y} - \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})\hat{\boldsymbol{\beta}}(\lambda) \\
&\quad - [\hat{\boldsymbol{\beta}}(\lambda)]^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})\boldsymbol{\beta} + \boldsymbol{\beta}^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})\boldsymbol{\beta} \\
&= \mathbf{Y}^\top\mathbf{Y} - \mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{Y} \\
&\quad + \left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\right]^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})\left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\right],
\end{aligned}
$$

the posterior distribution can be rewritten to:

$$
f_{\boldsymbol{\beta},\sigma^2}(\boldsymbol{\beta}, \sigma^2 \,|\, \mathbf{Y}, \mathbf{X}) \propto g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X})\, g_{\sigma^2}(\sigma^2 \,|\, \mathbf{Y}, \mathbf{X})
$$

with

$$
g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X}) \propto \exp\left\{ -\frac{1}{2\sigma^2}\left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\right]^\top(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})\left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\lambda)\right] \right\}.
$$

Then, clearly the conditional posterior mean of $\boldsymbol{\beta}$ is $\mathbb{E}(\boldsymbol{\beta} \,|\, \sigma^2, \mathbf{Y}, \mathbf{X}) = \hat{\boldsymbol{\beta}}(\lambda)$. Hence, the ridge regression estimator can be viewed as the Bayesian posterior mean estimator of $\boldsymbol{\beta}$ when imposing a Gaussian prior on the regression parameter.

With little extra work we may also obtain the conditional posterior of σ^2 from the joint posterior distribution:

$$f_{\sigma^2}(\sigma^2 \,|\, \boldsymbol{\beta}, \mathbf{Y}, \mathbf{X}) \quad \propto \quad (\sigma^2)^{-[(n+p)/2+\alpha_0+1]} \exp[-\frac{1}{2\sigma^2}(\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 + \beta_0)],$$

in which one can recognize the shape of an inverse gamma distribution.

A Bayesian estimator of a parameter $\boldsymbol{\theta}$ is the estimator that minimizes the Bayes risk over a prior distribution of the parameter $\boldsymbol{\theta}$. The Bayes risk is defined as $\int_{\boldsymbol{\theta}} \mathbb{E}[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^\top(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})]\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \boldsymbol{\alpha})d\boldsymbol{\theta}$, where $\pi_{\boldsymbol{\theta}}(\boldsymbol{\theta}; \boldsymbol{\alpha})$ is the prior distribution of $\boldsymbol{\theta}$ with hyperparameter $\boldsymbol{\alpha}$. It is thus a weighted average of the Mean Squared Error, with weights specified through the prior. The Bayes risk is minimized by the mean posterior $\mathbb{E}_{\boldsymbol{\theta}}(\boldsymbol{\theta} \,|\, \text{data})$ (cf., e.g., Bijma *et al.*, 2017). The Bayesian estimator of $\boldsymbol{\theta}$ thus yields the smallest possible expected MSE, under the assumption of the employed prior.

The Bayes risk of the ridge estimator over the normal prior $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}_p, \sigma^2\lambda^{-1}\mathbf{I}_{pp})$ is:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\beta}}\{\text{MSE}[\hat{\boldsymbol{\beta}}(\lambda)] \,|\, \sigma^2, \mathbf{Y}, \mathbf{X}\} &= \sigma^2 \operatorname{tr}\{\mathbf{W}_\lambda (\mathbf{X}^\top\mathbf{X})^{-1} \mathbf{W}_\lambda^\top\} + \mathbb{E}_{\boldsymbol{\beta}}[\boldsymbol{\beta}^\top (\mathbf{W}_\lambda - \mathbf{I}_{pp})^\top (\mathbf{W}_\lambda - \mathbf{I}_{pp})\, \boldsymbol{\beta}] \\
&= \sigma^2 \left\{\operatorname{tr}[\mathbf{W}_\lambda (\mathbf{X}^\top\mathbf{X})^{-1} \mathbf{W}_\lambda^\top] + \lambda^{-1}\operatorname{tr}[(\mathbf{W}_\lambda - \mathbf{I}_{pp})^\top (\mathbf{W}_\lambda - \mathbf{I}_{pp})]\right\} \\
&= \sigma^2 \sum_{j=1}^p (d_{jj}^2 + \lambda)^{-1},
\end{aligned}
$$

in which we have used *i)* the previously derived explicit expression (1.7) of the ridge estimator's MSE, *ii)* the expectation of the quadratic form of a multivariate random variable (Mathai and Provost, 1992), *iii)* the singular value decomposition of $\mathbf{X}$ with singular values d_{jj}, and *iv)* the fact that the trace of a square matrix equals the sum of its eigenvalues. As the ridge estimator coincides with the posterior mean, this is the minimal achievable MSE under a zero-centered normal prior with an uncorrelated and equivariant covariance matrix.

Above the Bayes risk of the ridge estimator factorizes with respect to σ^2 and λ. Hence, the larger the hyperparameter λ the lower the Bayes risk of the ridge estimator. In particular, its Bayes risk converges to zero as $\lambda \to \infty$. This can be understood as follows. The limit corresponds to an infinite precision of the prior, thus reducing the variance contribution to the MSE. Moreover, as the ridge estimator shrinks towards zero and the prior distribution of $\boldsymbol{\beta}$ has a zero mean, the bias too vanishes as $\lambda \to \infty$.

The calculation of the Bayes risk above relates the Bayesian and frequentist statements on the MSE of the ridge estimator. For the latter revisit Theorem 1.2 of Section 1.4.3, which states the existence of a λ such that the resulting ridge estimator has a superior MSE over that of the ML estimator. This result made no assumption on (the distribution of) $\boldsymbol{\beta}$. In fact, it can be viewed as a statement of the MSE conditional on $\boldsymbol{\beta}$. The Bayesian result integrates out the uncertainty – specified by the prior – in $\boldsymbol{\beta}$ from the (frequentist's) conditional MSE to arrive at the unconditional MSE.

1.7   Degrees of freedom

The degrees of freedom consumed by ridge regression is calculated. The degrees of freedom may be used in combination with an information criterion to decide on the value of the penalty parameter. Recall from ordinary regression that:

$$\widehat{\mathbf{Y}} \quad = \quad \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} \quad = \quad \mathbf{H}\mathbf{Y},$$

where $\mathbf{H}$ is the hat matrix. The degrees of freedom used in the regression is then equal to $\operatorname{tr}(\mathbf{H})$, the trace of $\mathbf{H}$. In particular, if $\mathbf{X}$ is of full rank, i.e. $\operatorname{rank}(\mathbf{X}) = p$, then $\operatorname{tr}(\mathbf{H}) = p$.

By analogy, the ridge-version of the hat matrix is:

$$\mathbf{H}(\lambda) \quad = \quad \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top.$$

Continuing this analogy, the degrees of freedom of ridge regression is given by the trace of the ridge hat matrix $\mathbf{H}(\lambda)$:

$$\operatorname{tr}[\mathbf{H}(\lambda)] \quad = \quad \operatorname{tr}[\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top] \quad = \quad \sum_{j=1}^p \frac{d_{jj}^2}{d_{jj}^2 + \lambda}.$$

The degrees of freedom consumed by ridge regression is monotone decreasing in λ. In particular:

$$\lim_{\lambda \to \infty} \text{tr}[\mathbf{H}(\lambda)] = 0.$$

That is, in the limit no information from $\mathbf{X}$ is used. Indeed, $\boldsymbol{\beta}$ is forced to equal $\mathbf{0}_p$ which is not derived from data.

1.8 Efficient calculation

In the high-dimensional setting the number of covariates p is large compared to the number of samples n. In a microarray experiment $p = 40000$ and $n = 100$ is not uncommon. To perform ridge regression in this context, the following expression needs to be evaluated numerically:

$$(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y}.$$

For $p = 40000$ this requires the inversion of a 40000×40000 dimensional matrix. This is not feasible on most desktop computers. However, there is a workaround.

Revisit the singular value decomposition of $\mathbf{X} = \mathbf{U}_x \mathbf{D}_x \mathbf{V}_x^\top$ and write $\mathbf{R}_x = \mathbf{U}_x \mathbf{D}_x$. As both $\mathbf{U}_x$ and $\mathbf{D}_x$ are $(n \times n)$-dimensional matrices, so is $\mathbf{R}_x$. Consequently, $\mathbf{X}$ is now decomposed as $\mathbf{X} = \mathbf{R}_x \mathbf{V}_x^\top$. The ridge estimator can be rewritten in terms of $\mathbf{R}_x$ and $\mathbf{V}_x$:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}(\lambda) &= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y} \\
&= (\mathbf{V}_x \mathbf{R}_x^\top \mathbf{R}_x \mathbf{V}_x^\top + \lambda \mathbf{I}_{pp})^{-1} \mathbf{V}_x \mathbf{R}_x^\top \mathbf{Y} \\
&= (\mathbf{V}_x \mathbf{R}_x^\top \mathbf{R}_x \mathbf{V}_x^\top + \lambda \mathbf{V}_x \mathbf{V}_x^\top)^{-1} \mathbf{V}_x \mathbf{R}_x^\top \mathbf{Y} \\
&= \mathbf{V}_x (\mathbf{R}_x^\top \mathbf{R}_x + \lambda \mathbf{I}_{nn})^{-1} \mathbf{V}_x^\top \mathbf{V}_x \mathbf{R}_x^\top \mathbf{Y} \\
&= \mathbf{V}_x (\mathbf{R}_x^\top \mathbf{R}_x + \lambda \mathbf{I}_{nn})^{-1} \mathbf{R}_x^\top \mathbf{Y}.
\end{aligned}
$$

Hence, the reformulated ridge estimator involves the inversion of an $(n \times n)$-dimensional matrix. With $n = 100$ this is feasible on most standard computers.

Hastie and Tibshirani (2004) point out that the number of computation operations reduces from $\mathcal{O}(p^3)$ to $\mathcal{O}(pn^2)$. In addition, they point out that this computational short-cut can be used in combination with other loss functions, for instance that of standard generalized linear models.

Avoidance of the inversion of the $(p \times p)$-dimensional matrix may be achieved in an other way. Hereto one needs the Woodbury identity. Let $\mathbf{A}$, $\mathbf{U}$ and $\mathbf{V}$ be $(p \times p)$-, $(p \times n)$- and $(n \times p)$-dimensional matrices, respectively. The (simplified form of the) Woodbury identity then is:

$$(\mathbf{A} + \mathbf{U}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I}_{nn} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1}.$$

Application of the Woodbury identity to the matrix inverse in the ridge estimator of the regression parameter gives:

$$(\lambda \mathbf{I}_{pp} + \mathbf{X}^\top \mathbf{X})^{-1} = \lambda^{-1}\mathbf{I}_{pp} - \lambda^{-2}\mathbf{X}^\top(\mathbf{I}_{nn} + \lambda^{-1}\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}.$$

This gives:

$$
\begin{aligned}
(\lambda \mathbf{I}_{pp} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} &= \lambda^{-1}\mathbf{X}^\top\mathbf{Y} - \lambda^{-2}\mathbf{X}^\top(\mathbf{I}_{nn} + \lambda^{-1}\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{X}^\top\mathbf{Y} \\
&= \lambda^{-1}\mathbf{X}^\top\left[\mathbf{Y} - \lambda^{-1}\mathbf{X}^\top(\mathbf{I}_{nn} + \lambda^{-1}\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{Y}\right].
\end{aligned}
$$

The inversion of the $(p \times p)$-dimensional matrix $\lambda \mathbf{I}_{pp} + \mathbf{X}^\top \mathbf{X}$ is thus replaced by that of the $(n \times n)$-dimensional matrix $\mathbf{I}_{nn} + \lambda^{-1}\mathbf{X}\mathbf{X}^\top$. In addition, this expression of the ridge regression estimator avoids the singular value decomposition of $\mathbf{X}$, which may in some cases introduce additional numerical errors (e.g. at the level of machine precision).

1.9 Choice of the penalty parameter

Throughout the introduction of ridge regression and the subsequent discussion of its properties the penalty parameter is considered known or 'given'. In practice, it is unknown and the user needs to make an informed decision on its value. Several strategies to facilitate such a decision are presented.

1.9.1 Information criterion

A popular strategy is to choose a penalty parameter that yields a good but parsimonious model. Information criteria measure the balance between model fit and model complexity. Here we present the Aikaike's information criterion (AIC), but many other criteria have been presented in the literature (e.g. Akaike, 1974, Schwarz, 1978). The AIC measures model fit by the log-likelihood and model complexity is measured by the number of parameters used by the model. The number of model parameters in regular regression simply corresponds to the number of covariates in the model. Or, by the degrees of freedom consumed by the model, which is equivalent to the trace of the hat matrix. For ridge regression it thus seems natural to define model complexity analogously by the trace of the ridge hat matrix. This yields the AIC for the linear regression model with ridge estimates:

$$
\begin{aligned}
\mathrm{AIC}(\lambda) &= 2\,p - 2\log(\hat{L}) \\
&= 2\,\mathrm{tr}[\mathbf{H}(\lambda)] - 2\log\{L[\hat{\boldsymbol{\beta}}(\lambda), \hat{\sigma}^2(\lambda)]\} \\
&= 2\sum_{j=1}^{p} \frac{d_{jj}^2}{d_{jj}^2 + \lambda} + 2n\,\log[\sqrt{2\,\pi}\,\hat{\sigma}(\lambda)] + \frac{1}{\hat{\sigma}^2(\lambda)}\sum_{i=1}^{n}[y_i - \mathbf{X}_{i,*}\,\hat{\boldsymbol{\beta}}(\lambda)]^2.
\end{aligned}
$$

The value of λ which minimizes $\mathrm{AIC}(\lambda)$ corresponds to the 'optimal' balance of model complexity and overfitting.

Information criteria guide the decision process when having to decide among various different models. Different models use different sets of explanatory variables to explain the behaviour of the response variable. In that sense, the use of information criteria for the deciding on the ridge penalty parameter may be considered inappropriate: ridge regression uses the same set of explanatory variables irrespective of the value of the penalty parameter. Moreover, often ridge regression is employed to predict a response and not to provide an insightful explanatory model. The latter need not yield the best predictions. Finally, empirically we observe that the AIC often does not show an optimum *inside* the domain of the ridge penalty parameter. Henceforth, we refrain from the use of the AIC (or any of its relatives) in determining the optimal ridge penalty parameter.

1.9.2 Cross-validation

Instead of choosing the penalty parameter to balance model fit with model complexity, cross-validation requires it (i.e. the penalty parameter) to yield a model with good prediction performance. Commonly, this performance is evaluated on novel data. Novel data need not be easy to come by and one has to make do with the data at hand. The setting of 'original' and novel data is then mimicked by sample splitting: the data set is divided into two (groups of samples). One of these two data sets, called the *training set*, plays the role of 'original' data on which the model is built. The second of these data sets, called the *test set*, plays the role of the 'novel' data and is used to evaluate the prediction performance (often operationalized as the log-likelihood or the prediction error) of the model built on the training data set. This procedure (model building and prediction evaluation on training and test set, respectively) is done for a collection of possible penalty parameter choices. The penalty parameter that yields the model with the best prediction performance is to be preferred. The thus obtained performance evaluation depends on the actual split of the data set. To remove this dependence the data set is split many times into a training and test set. For each split the model parameters are estimated for all choices of λ using the training data and estimated parameters are evaluated on the corresponding test set. The penalty parameter that on average over the test sets performs best (in some sense) is then selected.

When the repetitive splitting of the data set is done randomly, samples may accidently end up in a fast majority of the splits in either training or test set. Such samples may have an unbalanced influence on either model building or prediction evaluation. To avoid this k-fold cross-validation structures the data splitting. The samples are divided into k more or less equally sized exhaustive and mutually exclusive subsets. In turn (at each split) one of these subsets plays the role of the test set while the union of the remaining subsets constitutes the training set. Such a splitting warrants a balanced representation of each sample in both training and test set over the splits. Still the division into the k subsets involves a degree of randomness. This may be fully excluded when choosing $k = n$. This particular case is referred to as leave-one-out cross-validation (LOOCV). For illustration purposes the LOOCV procedure is detailed fully below:

0) Define a range of interest for the penalty parameter.

1) Divide the data set into training and test set comprising samples $\{1, \ldots, n\} \setminus i$ and $\{i\}$, respectively.
2) Fit the linear regression model by means of ridge estimation for each λ in the grid using the training set. This yields:

$$\hat{\boldsymbol{\beta}}_{-i}(\lambda) = (\mathbf{X}_{-i,*}^\top \mathbf{X}_{-i,*} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}_{-i,*}^\top \mathbf{Y}_{-i}$$

and the corresponding estimate of the error variance $\hat{\sigma}_{-i}^2(\lambda)$.
3) Evaluate the prediction performance of these models on the test set by $\log\{L[Y_i, \mathbf{X}_{i,*}; \hat{\boldsymbol{\beta}}_{-i}(\lambda), \hat{\sigma}_{-i}^2(\lambda)]\}$.
Or, by the prediction error $|Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}_{-i}(\lambda)|$, possibly squared.
4) Repeat steps 1) to 3) such that each sample plays the role of the test set once.
5) Average the prediction performances of the test sets at each grid point of the penalty parameter:

$$\frac{1}{n}\sum_{i=1}^{n} \log\{L[Y_i, \mathbf{X}_{i,*}; \hat{\boldsymbol{\beta}}_{-i}(\lambda), \hat{\sigma}_{-i}^2(\lambda)]\}.$$

The quantity above is called the *cross-validated log-likelihood*. It is an estimate of the prediction performance of the model corresponding to this value of the penalty parameter on novel data.
6) The value of the penalty parameter that maximizes the cross-validated log-likelihood is the value of choice.

The procedure is straightforwardly adopted to k-fold cross-validation, a different criterion, and different estimators.

In the LOOCV procedure above resampling can be avoided when the prediction performance is measured by Allen's PRESS (Predicted Residual Error Sum of Squares) statistic (Allen, 1974). For then, the LOOCV prediction performance can be expressed analytically in terms of the known quantities derived from the design matrix and response (as pointed out but not detailed in Golub *et al.* 1979). Define the optimal penalty parameter to minimize Allen's PRESS statistic:

$$\lambda_{\text{opt}} = \arg\min_{\lambda} \frac{1}{n}\sum_{i=1}^{n}[Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}_{-i}(\lambda)]^2.$$

To derive an analytic expression for the right-hand side first rewrite $(\mathbf{X}_{-i,*}^\top \mathbf{X}_{-i,*} + \lambda \mathbf{I}_{pp})^{-1}$ by means of the Woodbury identity as:

$$\begin{aligned}
(\mathbf{X}_{-i,*}^\top \mathbf{X}_{-i,*} + \lambda \mathbf{I}_{pp})^{-1} &= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp} - \mathbf{X}_{i,*}^\top \mathbf{X}_{i,*})^{-1} \\
&= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} + (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{X}_{i,*}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top]^{-1} \\
&\qquad \mathbf{X}_{i,*}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \\
&= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} + (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}\mathbf{X}_{i,*}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}
\end{aligned}$$

with $\mathbf{H}_{ii}(\lambda) = \mathbf{X}_{i,*}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top$. Furthermore, $\mathbf{X}_{-i}^\top \mathbf{Y}_{-i} = \mathbf{X}^\top \mathbf{Y} - \mathbf{X}_{i,*}^\top Y_i$. Substitute both in the leave-one-out ridge regression estimator and manipulate:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{-i}(\lambda) &= (\mathbf{X}_{-i,*}^\top \mathbf{X}_{-i,*} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{-i,*}^\top \mathbf{Y}_{-i} \\
&= \{(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} + (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}\mathbf{X}_{i,*}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\} \\
&\quad \times(\mathbf{X}^\top \mathbf{Y} - \mathbf{X}_{i,*}^\top Y_i) \\
&= (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}^\top \mathbf{Y} - (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top Y_i \\
&\quad + (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}\mathbf{X}_{i,*}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}^\top \mathbf{Y} \\
&\quad - (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}\mathbf{X}_{i,*}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top Y_i \\
&= \hat{\boldsymbol{\beta}}(\lambda) - (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}[1 - \mathbf{H}_{ii}(\lambda)]Y_i \\
&\quad + (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}\mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda) \\
&\quad - (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}\mathbf{H}_{ii}(\lambda)Y_i \\
&= \hat{\boldsymbol{\beta}}(\lambda) - (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}\{[1 - \mathbf{H}_{ii}(\lambda)]Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda) + \mathbf{H}_{ii}(\lambda)Y_i\} \\
&= \hat{\boldsymbol{\beta}}(\lambda) - (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}[Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda)].
\end{aligned}$$

The latter enables the reformulation of the prediction error as:

$$
\begin{aligned}
Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}_{-i}(\lambda) &= Y_i - \mathbf{X}_{i,*}\{\hat{\boldsymbol{\beta}}(\lambda) - (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}[Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda)]\} \\
&= Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda) + \mathbf{X}_{i,*}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}_{i,*}^\top[1 - \mathbf{H}_{ii}(\lambda)]^{-1}[Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda)] \\
&= Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda) + \mathbf{H}_{ii}(\lambda)[1 - \mathbf{H}_{ii}(\lambda)]^{-1}[Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}(\lambda)] \\
&= [1 - \mathbf{H}_{ii}(\lambda)]^{-1}[Y_i - \mathbf{X}_{i,*}^\top\hat{\boldsymbol{\beta}}(\lambda)],
\end{aligned}
$$

which in turn results in the re-expression of Allen's PRESS statistic:

$$
\lambda_{\text{opt}} = \arg\min_\lambda \frac{1}{n}\sum_{i=1}^n [Y_i - \mathbf{X}_{i,*}\hat{\boldsymbol{\beta}}_{-i}(\lambda)]^2 = \arg\min_\lambda \frac{1}{n}\|\mathbf{B}(\lambda)[\mathbf{I}_{nn} - \mathbf{H}(\lambda)]\mathbf{Y}\|_F^2,
$$

where $\mathbf{B}(\lambda)$ is diagonal with $[\mathbf{B}(\lambda)]_{ii} = [1 - \mathbf{H}_{ii}(\lambda)]^{-1}$. Hence, the prediction performance for a given λ can be assessed directly from the ridge hat matrix and the response vector without the recalculation of the n leave-one-out ridge estimators. Computationally, this is a considerable gain.

1.10 Simulations

Simulations are presented that illustrate properties of the ridge estimator not discussed explicitly in the previous sections of this chapter.

1.10.1 Role of the variance of the covariates

In many applications of high-dimensional data the covariates are standardized prior to the execution of the ridge regression. Before we discuss whether this is appropriate, we first illustrate the effect of ridge penalization on covariates with distinct variances using simulated data.

The simulation involves one response to be (ridge) regressed on fifty covariates. Data (with $n = 1000$) for the covariates, denoted $\mathbf{X}$, are drawn from a multivariate normal distribution: $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_{50}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ diagonal and $(\boldsymbol{\Sigma})_{jj} = j/10$. From this the response is generated through $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\beta} = \mathbf{1}_{50}$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_{50}, \mathbf{I}_{50\times50})$.

With the simulated data at hand the ridge regression estimates of $\boldsymbol{\beta}$ are evaluated for a large grid of the penalty parameter λ. The resulting ridge regularization paths of the regression coefficients are plotted (Figure 1.6). All paths start ($\lambda = 0$) close to one and vanish as $\lambda \to \infty$. However, ridge regularization paths of regression coefficients corresponding to covariates with a large variance dominate those with a low variance.

Ridge regression's preference of covariates with a large variance can intuitively be understood as follows. First note that the ridge regression estimator now can be written as:

$$
\begin{aligned}
\boldsymbol{\beta}(\lambda) &= [\text{Var}(\mathbf{X}) + \lambda\mathbf{I}_{50\times50}]^{-1}\text{Cov}(\mathbf{X}, \mathbf{Y}) \\
&= (\boldsymbol{\Sigma} + \lambda\mathbf{I}_{50\times50})^{-1}\boldsymbol{\Sigma}[\text{Var}(\mathbf{X})]^{-1}\text{Cov}(\mathbf{X}, \mathbf{Y}) \\
&= (\boldsymbol{\Sigma} + \lambda\mathbf{I}_{50\times50})^{-1}\boldsymbol{\Sigma}\boldsymbol{\beta}.
\end{aligned}
$$

Plug in the employed parametrization of $\boldsymbol{\Sigma}$, which gives:

$$
[\boldsymbol{\beta}(\lambda)]_j = \frac{j}{j + 50\lambda}(\boldsymbol{\beta})_j.
$$

Hence, the larger the covariate's variance (corresponding to the larger j), the larger its ridge regression coefficient estimate. Ridge regression thus prefers (among a set of covariates with comparable effect sizes) those with larger variances.

The reformulation of ridge penalized estimation as a constrained estimation problem offers a geometrical interpretation of this phenomenon. Let $p = 2$ and the design matrix $\mathbf{X}$ be orthogonal, while both covariates contribute equally to the response. Contrast the cases with $\text{Var}(X_1) \approx \text{Var}(X_2)$ and $\text{Var}(X_1) \gg \text{Var}(X_2)$. The level sets of the least squares loss function associated with the former case are circular, while that of the latter are strongly ellipsoidal (see Figure 1.6). The diameters along the principal axes (that – due to the orthogonality of $\mathbf{X}$ – are parallel to that of the β_1- and β_2-axes) of

Solution paths of covariates with distinct variance

Ridge estimates with equal covariate variances

Ridge estimates with unequal covariate variances
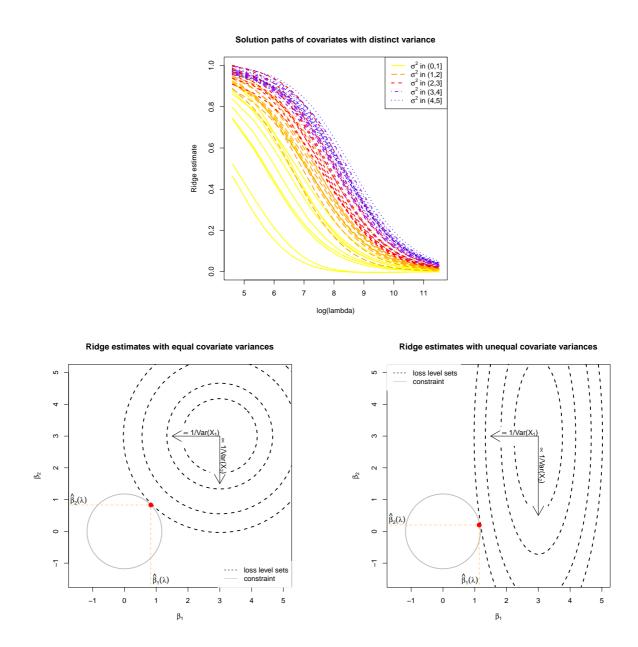

Figure 1.6: Top panel: Ridge regularization paths for coefficients of the 50 uncorrelated covariates with distinct variances. Color and line type indicated the grouping of the covariates by their variance. Bottom panels: Graphical illustration of the effect of a covariate's variance on the ridge estimator. The grey circle depicts the ridge parameter constraint. The dashed black ellipsoids are the level sets of the least squares loss function. The red dot is the ridge regression estimate. Left and right panels represent the cases with equal and unequal, respectively, variances of the covariates.

both circle and ellipsoid are reciprocals of the variance of the covariates. When the variances of both covariates are equal, the level sets of the loss function expand equally fast along both axis. With the two covariates having the same regression coefficient, the point of these level sets closest to the parameter constraint is to be found on the line $\beta_1 = \beta_2$ (Figure 1.6, left panel). Consequently, the ridge regression estimate satisfies $\hat{\beta}_1(\lambda) \approx \hat{\beta}_2(\lambda)$. With unequal variances between the covariates, the ellipsoidal level sets of the loss function have diameters of rather different sizes. In particular, along the β_1-axis it is narrow (as $\text{Var}(X_1)$ is large), and – vice versa – wide along the β_2-axis. Consequently, the point of these level sets closest to the circular parameter constraint will be closer to the β_1- than to the β_2-axis (Figure 1.6, left panel). For the ridge estimates of the regression parameter this implies $0 \ll \hat{\beta}_1(\lambda) < 1$ and $0 < \hat{\beta}_2(\lambda) \ll 1$. Hence, the covariate with a larger variance yields the larger ridge regression estimate.

Should one thus standardize the covariates prior to ridge regression analysis? When dealing with gene expression data from microarrays, the data have been subjected to a series of pre-processing steps (e.g. quality control, background correction, within- and between-normalization). The purpose of these steps is to make the expression levels of genes comparable both within and between hybridizations. The preprocessing should thus be considered an inherent part of the measurement. As such it is to be done independently of whatever down-stream analysis is to follow and further tinkering with the data is preferably to be avoided (as it may mess up the 'comparable-ness' of the expression levels as achieved by the preprocessing). For other data types different considerations may apply.

Among the considerations to decide on standardization of the covariates, one should also include the fact that ridge estimates prior and posterior to scaling do not simply differ by a factor. To see this assume that the covariates have been centered. Scaling of the covariates amounts to post-multiplication of the design matrix by a $(p \times p)$-dimensional diagonal matrix $\mathbf{A}$ with the reciprocals of the covariates' scale estimates on its diagonal (Sardy, 2008). Hence, the ridge estimator (for the rescaled data) is then given by:

$$\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\mathbf{A}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2.$$

Apply the change-of-variable $\boldsymbol{\gamma} = \mathbf{A}\boldsymbol{\beta}$ and obtain:

$$\min_{\boldsymbol{\gamma}} \|\mathbf{Y} - \mathbf{X}\gamma\|_2^2 + \lambda\|\mathbf{A}^{-1}\boldsymbol{\gamma}\|_2^2 \quad = \quad \min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\gamma\|_2^2 + \sum_{j=1}^{p} \lambda[(\mathbf{A})_{jj}]^{-2}\gamma_j^2.$$

Effectively, the scaling is equivalent to covariate-wise penalization. The 'scaled' ridge estimator may then be derived along the same lines as before in Section 1.5:

$$\hat{\boldsymbol{\beta}}^{(\text{scaled})}(\lambda) \quad = \quad \mathbf{A}^{-1}\hat{\boldsymbol{\gamma}}(\lambda) \quad = \quad \mathbf{A}^{-1}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{A}^{-2})^{-1}\mathbf{X}^\top\mathbf{Y}.$$

In general, this is unequal to the ridge estimator without the rescaling of the columns of the design matrix. Moreover, it should be clear that $\hat{\boldsymbol{\beta}}^{(\text{scaled})}(\lambda) \neq \mathbf{A}\hat{\boldsymbol{\beta}}(\lambda)$.

1.10.2 Ridge regression and collinearity

Initially, ridge regression was motivated as an ad-hoc fix of (super)-collinear covariates in order to obtain a well-defined estimator. We now study the effect of this ad-hoc fix on the regression coefficient estimates of collinear covariates. In particular, their ridge regularization paths are contrasted to those of 'non-collinear' covariates.

To this end, we consider a simulation in which one response is regressed on 50 covariates. The data of these covariates, stored in a design matrix denoted $\mathbf{X}$, are sampled from a multivariate normal distribution, with mean zero and a 5×5 blocked covariance matrix:

$$\boldsymbol{\Sigma} \quad = \quad \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} \\ \mathbf{0}_{10 \times 10} & \boldsymbol{\Sigma}_{22} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} \\ \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \boldsymbol{\Sigma}_{33} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} \\ \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \boldsymbol{\Sigma}_{44} & \mathbf{0}_{10 \times 10} \\ \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \mathbf{0}_{10 \times 10} & \boldsymbol{\Sigma}_{55} \end{pmatrix}$$

with

$$\boldsymbol{\Sigma}_{kk} \quad = \quad \frac{k-1}{5}\mathbf{1}_{10 \times 10} + \frac{6-k}{5}\mathbf{I}_{10 \times 10}.$$

The data of the response variable $\mathbf{Y}$ are then obtained through: $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_n, \mathbf{I}_{nn})$ and $\boldsymbol{\beta} = \mathbf{1}_{50}$. Hence, all covariates contribute equally to the response. Would the columns of $\mathbf{X}$ be orthogonal, little difference in the ridge estimates of the regression coefficients is expected.

The results of this simulation study with sample size $n = 1000$ are presented in Figure 1.7. All 50 regularization paths start close to one as λ is small and converge to zero as $\lambda \to \infty$. But the paths of covariates of the same block of the covariance matrix $\boldsymbol{\Sigma}$ quickly group, with those corresponding to a block with larger off-diagonal elements above those with smaller ones. Thus, ridge regression prefers (i.e. shrinks less) coefficient estimates of strongly positively correlated covariates.
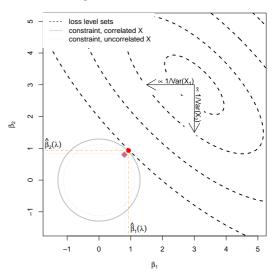
Figure 1.7: Left panel: Ridge regularization paths for coefficients of the 50 covariates, with various degree of collinearity but equal variance. Color and line type correspond to the five blocks of the covariate matrix Σ. Right panel: Graphical illustration of the effect of the collinearity among covariates on the ridge estimator. The solid and dotted grey circles depict the ridge parameter constraint for the collinear and orthogonal cases, respectively. The dashed black ellipsoids are the level sets of the sum-of-squares squares loss function. The red dot and violet diamond are the ridge regression for the positive collinear and orthogonal case, respectively.

Intuitive understanding of the observed behaviour may be obtained from the $p = 2$ case. Let U, V and ε be independent random variables with zero mean. Define $X_1 = U + V$, $X_2 = U - V$, and $Y = \beta_1 X_1 + \beta_2 X_2 + \varepsilon$ with β_1 and β_2 constants. Hence, $\mathbb{E}(Y) = 0$. Then:

$$
\begin{aligned}
Y &= (\beta_1 + \beta_2)U + (\beta_1 - \beta_2)V + \varepsilon \\
&= \gamma_u U + \gamma_v V + \varepsilon
\end{aligned}
$$

and $\mathrm{Cor}(X_1, X_2) = [\mathrm{Var}(U) - \mathrm{Var}(V)]/[\mathrm{Var}(U) + \mathrm{Var}(V)]$. The random variables X_1 and X_2 are strongly positively correlated if $\mathrm{Var}(U) \gg \mathrm{Var}(V)$.

The ridge regression estimator associated with regression of Y on U and V is:

$$
\boldsymbol{\gamma}(\lambda) = \begin{pmatrix} \mathrm{Var}(U) + \lambda & 0 \\ 0 & \mathrm{Var}(V) + \lambda \end{pmatrix}^{-1} \begin{pmatrix} \mathrm{Cov}(U, Y) \\ \mathrm{Cov}(V, Y) \end{pmatrix}.
$$

For large enough λ

$$
\boldsymbol{\gamma}(\lambda) \approx \frac{1}{\lambda} \begin{pmatrix} \mathrm{Var}(U) & 0 \\ 0 & \mathrm{Var}(V) \end{pmatrix} \begin{pmatrix} \beta_1 + \beta_2 \\ \beta_1 - \beta_2 \end{pmatrix}.
$$

When $\mathrm{Var}(U) \gg \mathrm{Var}(V)$ and $\beta_1 \approx \beta_2$, the ridge estimate of γ_v vanishes for large λ. Hence, ridge regression prefers positively covariates with similar effect sizes.

This phenomenon too can be explained geometrically. For the illustration consider ridge estimation with $\lambda = 1$ of the linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\beta} = (3, 3)^\top$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_2, \mathbf{I}_{22})$ and the columns of $\mathbf{X}$ strongly and positively collinear. The level sets of the sum-of-squares loss, $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$, are plotted in the right panel of Figure 1.7. Recall that the ridge estimate is found by looking for the smallest loss level set that hits the ridge contraint. The sought-for estimate is then the point of intersection between this level set and the constraint, and – for the case at hand – is on the $x = y$-line. This is no different from the case with orthogonal $\mathbf{X}$ columns. Yet their estimates differ, even though the same λ is applied. The difference is to due to fact that the radius of the ridge constraint depends on λ, $\mathbf{X}$ and $\mathbf{Y}$. This is immediate from the fact that the radius of the constraint equals $\|\hat{\boldsymbol{\beta}}(\lambda)\|_2^2$ (see Section 1.5). To study the

effect of $\mathbf{X}$ on the radius, we remove its dependence on $\mathbf{Y}$ by considering its expectation, which is:

$$
\begin{aligned}
\mathbb{E}[\|\hat{\boldsymbol{\beta}}(\lambda)\|_2^2] &= \mathbb{E}\{[(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top\mathbf{X})\,\hat{\boldsymbol{\beta}}]^\top\,(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_{pp})^{-1}(\mathbf{X}^\top\mathbf{X})\,\hat{\boldsymbol{\beta}}\} \\
&= \mathbb{E}[\mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_{pp})^{-2}\mathbf{X}^\top\mathbf{Y}] \\
&= \sigma^2\,\mathrm{tr}\{\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_{pp})^{-2}\mathbf{X}^\top\} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X}+\lambda\mathbf{I}_{pp})^{-2}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta}.
\end{aligned}
$$

In the last step we have used $\mathbf{Y}\sim\mathcal{N}(\mathbf{X}\boldsymbol{\beta},\sigma^2\mathbf{I}_{pp})$ and the expectation of the quadratic form of a multivariate random variable $\boldsymbol{\varepsilon}\sim\mathcal{N}(\boldsymbol{\mu}_\varepsilon,\boldsymbol{\Sigma}_\varepsilon)$ is $\mathbb{E}(\boldsymbol{\varepsilon}^\top\boldsymbol{\Lambda}\,\boldsymbol{\varepsilon}) = \mathrm{tr}(\boldsymbol{\Lambda}\,\boldsymbol{\Sigma}_\varepsilon)+\boldsymbol{\mu}_\varepsilon^\top\boldsymbol{\Lambda}\,\boldsymbol{\mu}_\varepsilon$ (cf. Mathai and Provost, 1992). The expression for the expectation of the radius of the ridge constraint can now be evaluated for the orthogonal $\mathbf{X}$ and the strongly, positively collinear $\mathbf{X}$. It turns out that the latter is larger than the former. This results in a larger ridge constraint. For the larger ridge constraint there is a smaller level set that hits it first. The point of intersection, still on the $x = y$-line, is now thus closer to $\boldsymbol{\beta}$ and further from the origin (cf. right panel of Figure 1.7). The resulting estimate is thus larger than that from the orthogonal case.

The above needs some attenuation. Among others it depends on: *i)* the number of covariates in each block, *ii)* the size of the effects, i.e. regression coefficients of each covariate, and *iii)* the degree of collinearity. Possibly, there are more factors influencing the behaviour of the ridge estimator presented in this subsection.

This behaviour of ridge regression is to be understood when using (say) gene expression data to predict a certain clinical outcome. Genes work in concert to fulfil a certain function in the cell. Consequently, one expects their expression levels to be correlated. Indeed, gene expression studies exhibit many co-expressed genes, that is, genes with correlating transcript levels.

1.11   Illustration

The application of ridge regression to actual data aims to illustrate its use in practice.

1.11.1   MCM7 expression regulation by microRNAs

Recently, a new class of RNA was discovered, referred to as microRNA. MicroRNAs are non-coding, single stranded RNAs of approximately 22 nucleotides. Like mRNAs, microRNAs are encoded in and transcribed from the DNA. MicroRNAs play an important role in the regulatory mechanism of the cell. MicroRNAs down-regulate gene expression by either of two post-transcriptional mechanisms: mRNA cleavage or transcriptional repression. This depends on the degree of complementarity between the microRNA and the target. Perfect or nearly perfect complementarity of the mRNA to the microRNA will lead to cleavage and degradation of the target mRNA. Imperfect complementarity will repress the productive translation and reduction in protein levels without affecting the mRNA levels. A single microRNA can bind to and regulate many different mRNA targets. Conversely, several microRNAs can bind to and cooperatively control a single mRNA target (Bartel, 2004; Esquela-Kerscher and Slack, 2006; Kim and Nam, 2006).

In this illustration we wish to confirm the regulation of mRNA expression by microRNAs in an independent data set. We cherry pick an arbitrary finding from literature reported in Ambs *et al.* (2008), which focusses on the microRNA regulation of the MCM7 gene in prostate cancer. The MCM7 gene is involved in DNA replication (Tye, 1999), a cellular process often derailed in cancer. Furthermore, MCM7 interacts with the tumor-suppressor gene RB1 (Sterner *et al.*, 1998). Several studies indeed confirm the involvement of MCM7 in prostate cancer (Padmanabhan *et al.*, 2004). And recently, it has been reported that in prostate cancer MCM7 may be regulated by microRNAs (Ambs *et al.*, 2008).

We here assess whether the MCM7 down-regulation by microRNAs can be observed in a data set other than the one upon which the microRNA-regulation of MCM7 claim has been based. To this end we download from the Gene Expression Omnibus (GEO) a prostate cancer data set (presented by Wang *et al.*, 2009). This data set (with GEO identifier: GSE20161) has both mRNA and microRNA profiles for all samples available. The preprocessed (as detailed in Wang *et al.*, 2009) data are downloaded and require only minor further manipulations to suit our purpose. These manipulations comprise *i)* averaging of duplicated profiles of several samples, *ii)* gene- and mir-wise zero-centering of the expression data, *iii)* averaging the expression levels of the probes that interrogate MCM7. Eventually, this leaves 90 profiles each comprising of 735 microRNA expression measurements.

Listing 1.2 R code

```r
# load libraries
library(GEOquery)
library(RmiR.hsa)
library(penalized)

# extract data
slh      <- getGEO("GSE20161", GSEMatrix=TRUE)
GEdata  <- slh[1][[1]]
MIRdata <- slh[2][[1]]

# average duplicate profiles
Yge  <- numeric()
Xmir <- numeric()
for (sName in 1:90){
  Yge  <- cbind(Yge,  apply(exprs(GEdata)[,sName,drop=FALSE],  1, mean))
  Xmir <- cbind(Xmir, apply(exprs(MIRdata)[,sName,drop=FALSE], 1, mean))
}
colnames(Yge)  <- paste("S", 1:90, sep="")
colnames(Xmir) <- paste("S", 1:90, sep="")

# extact mRNA expression of the MCM7N tumor suppressor gene
entrezID <- c("4176")
geneName <- "MCM7"
Y        <- Yge[which(levels(fData(GEdata)[,6])[fData(GEdata)[,6]] == geneName)
   ,]

# average gene expression levels over probes
Y <- apply(Y, 2, mean)

# mir-wise centering mir expression data
X <- t(sweep(Xmir, 1, rowMeans(Xmir)))

# generate cross-validated likelihood profile
profL2(Y, penalized=X, minlambda2=1, maxlambda2=20000, plot=TRUE)

# decide on the optimal penalty value directly
optLambda <- optL2(Y, penalized=X)$lambda

# obtain the ridge regression estimages
ridgeFit <- penalized(Y, penalized=X, lambda2=optLambda)

# plot them as histogram
hist(coef(ridgeFit, "penalized"), n=50, col="blue", border="lightblue",
     xlab="ridge regression estimates with optimal lambda",
     main="Histogram of ridge estimates")

#  linear prediction from ridge
Yhat <- predict(ridgeFit, X)[,1]
plot(Y ~ Yhat, pch=20, xlab="pred. MCM7 expression",
                       ylab="obs. MCM7 expression")
```

With this prostate data set at hand we now investigate whether MCM7 is regulated by microRNAs. Hereto we fit a linear regression model regressing the expression levels of MCM7 onto those of the microRNAs. As the number of microRNAs exceeds the number of samples, ordinary least squares fails and we resort to the ridge estimator of the regression coefficients. First, an informed choice of the penalty parameter is made through maximization of the LOOCV log-likelihood, resulting in $\lambda_{\mathrm{opt}} = 1812.826$. Having decided on the value of the to-be-employed penalty parameter, the ridge regression estimator can now readily be evaluated. The thus fitted model allows for the evaluation of microRNA-regulation of MCM7. E.g., by the proportion of variation of the MCM7 expression levels by the microRNAs as expressed in coefficient of determination: $R^2 = 0.4492$. Alternatively, but closely related, observed

expression levels may be related to the linear predictor of the MCM7 expression levels: $\hat{\mathbf{Y}}(\lambda_{\text{opt}}) = \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_{\text{opt}})$. The Spearman correlation of response and predictor equals 0.6295. A visual inspection is provided by the left panel of Figure 1.8. Note the difference in scale of the x- and y-axes. This is due to the fact that the regression coefficients have been estimated in penalized fashion, consequently shrinking estimates of the regression coefficients towards zero leading to small estimates and in turn compressing the range of the linear prediction. The above suggests there is indeed association between the microRNA expression levels and those of MCM7.
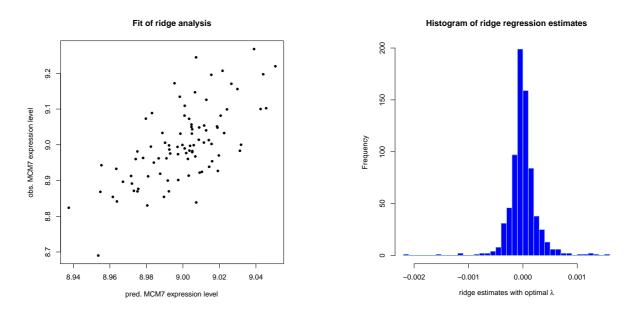

Figure 1.8: Left panel: Observed vs. (ridge) fitted MCM7 expression values. Right panel: Histogram of the ridge regression coefficient estimates.

The overall aim of this illustration was to assess whether microRNA-regulation of MCM7 could also be observed in this prostate cancer data set. In this endeavour the dogma (stating this regulation should be negative) has nowhere been used. A first simple assessment of the validity of this dogma studies the signs of the estimated regression coefficients. The ridge regression estimate has 394 out of the 735 microRNA probes with a negative coefficient. Hence, a small majority has a sign in line with the 'microRNA ↓ mRNA' dogma. When, in addition, taking the size of these coefficients into account (Figure 1.8, right panel), the negative regression coefficient estimates do not substantially differ from their positive counterparts (as can be witnessed from their almost symmetrical distribution around zero). Hence, the value of the 'microRNA ↓ mRNA' dogma is not confirmed by this ridge regression analysis of the MCM7-regulation by microRNAs. Nor is it refuted.

The implementation of ridge regression in the `penalized`-package offers the possibility to fully obey the dogma on negative regulation of mRNA expression by microRNAs. This requires all regression coefficients to be negative. Incorporation of the requirement into the ridge estimation augments the constrained estimation problem with an additional constraint:
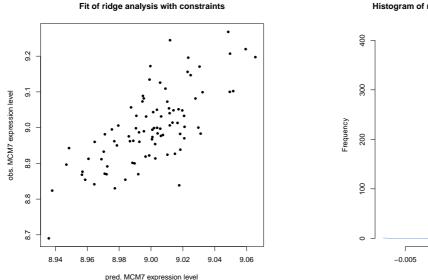
$$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\substack{\|\boldsymbol{\beta}\|_2^2 \leq c(\lambda) \\ \beta_j \leq 0 \text{ for all } j}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2.$$

With the additional non-positivity constraint on the parameters, there is no explicit solution for the estimator. The ridge estimate of the regression parameters is then found by numerical optimization using e.g. the Newton-Raphson algorithm or a gradient descent approach. The next listing gives the R-code for ridge estimation with the non-positivity constraint of the linear regression model.

Listing 1.3 R code

```
# decide on the optimal penalty value with sign constraint on paramers
optLambda <- optL2(Y, penalized=-X, positive=rep(TRUE, ncol(X)))$lambda
```

```
# obtain the ridge regression estimages
ridgeFit <- penalized(Y, penalized=-X, lambda2=optLambda,
                      positive=rep(TRUE, ncol(X)))

# linear prediction from ridge
Yhat <- predict(ridgeFit, -X)[,1]
plot(Y ~ Yhat, pch=20, xlab="predicted MCM7 expression level",
                       ylab="observed MCM7 expression level")
cor(Y, Yhat, m="s")
summary(lm(Y ~ Yhat))[8]
```

The linear regression model linking MCM7 expression to that of the microRNAs is fitted by ridge regression while simultaneously obeying the 'negative regulation of mRNA by microRNA'-dogma to the prostate cancer data. In the resulting model 401 out of 735 microRNA probes have a nonzero (and negative) coefficient. There is a large overlap in microRNAs with a negative coefficient between those from this and the previous fit. The models are also compared in terms of their fit to the data. The Spearman rank correlation coefficient between response and predictor for the model without positive regression coefficients equals 0.679 and its coefficient of determination 0.524 (confer the left panel of 1.9 for a visualization). This is a slight improvement upon the unconstrained ridge estimated model. The improvement may be small but it should be kept in mind that the number of parameters used by both models is 401 (for the model without positive regression coefficients) vs. 735. Hence, with close to half the number of parameters the dogma-obeying model gives a somewhat better description of the data. This may suggest that there is some value in the dogma as inclusion of this prior information leads to a more parsimonious model without any loss in fit.
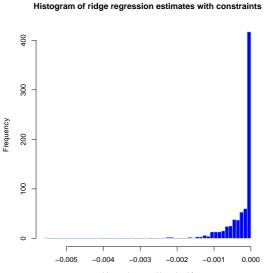

Figure 1.9: Left panel: Observed vs. (ridge) fitted MCM7 expression values (with the non-positive constraint on the parameters in place). Right panel: Histogram of the ridge regression coefficient estimates (from the non-positivity constrained analysis).

The dogma-obeying model selects 401 microRNAs that aid in the explanation of the variation in the gene expression levels of MCM7. There is an active field of research, called *target prediction*, trying to identify which microRNAs target the mRNA of which genes. Within R there is a collection of packages that provide the target prediction of known microRNAs. The packages differ on the method (e.g. experimental or sequence comparison) that has been used to arrive at the prediction. These target predictions may be used to evaluate the value of the found 401 microRNAs. Ideally, there would be a substantial amount of overlap. The R-script that loads the target predictions and does the comparison is below.

Listing 1.4 R code

```r
# extract mir names and their (hypothesized) mrna target
mir2target      <- numeric()
mirPredProgram <- c("targetscan", "miranda", "mirbase", "pictar", "mirtarget2")
for (program in mirPredProgram){
    slh            <- dbReadTable(RmiR.hsa_dbconn(), program)
    slh            <- cbind(program, slh[,1:2])
    colnames(slh) <- c("method", "mir", "target")
    mir2target     <- rbind(mir2target, slh)
}
mir2target <- unique(mir2target)
mir2target <- mir2target[which(mir2target[,3] == entrezID),]
uniqMirs   <- tolower(unique(mir2target[,2]))

# extract names of mir-probe on array
arrayMirs <- tolower(levels(fData(MIRdata)[,3])[fData(MIRdata)[,3]])

# which mir-probes are predicted to down-regulate MCM7
selMirs <- intersect(arrayMirs, uniqMirs)
ids     <- which(arrayMirs %in% selMirs)

# which ridge estimates are non-zero
nonzeroBetas <- (coef(ridgeFit, "penalized") != 0)

# which mirs are predicted to
nonzeroPred      <- 0 * betas
nonzeroPred[ids] <- 1

# contingency table and chi-square test
table(nonzeroBetas, nonzeroPred)
chisq.test(table(nonzeroBetas, nonzeroPred))
```

	$\hat{\beta}_j = 0$	$\hat{\beta}_j < 0$
microRNA not target	323	390
microRNA target	11	11

Table 1.1: Cross-tabulation of the microRNAs being a potential target of MCM7 vs. the value of its regression coefficient in the dogma-obeying model.

With knowledge available on each microRNA whether it is predicted (by at least one target prediction package) to be a potential target of MCM7, it may be cross-tabulated against its corresponding regression coefficient estimate in the dogma-obeying model being equal to zero or not. Table 1.1 contains the result. Somewhat superfluous considering the data, we may test whether the targets of MCM7 are overrepresented in the group of strictly negatively estimated regression coefficients. The corresponding chi-squared test (with Yates' continuity correction) yields the test statistic $\chi^2 = 0.0478$ with a p-value equal to 0.827. Hence, there is no enrichment among the 401 microRNAS of those that have been predicted to target MCM7. This may seem worrisome. However, the microRNAs have been selected for their predictive power of the expression levels of MCM7. Variable selection has not been a criterion (although the sign constraint implies selection). Moreover, criticism on the value of the microRNA target prediction has been accumulating in recent years.

1.12   Conclusion

We discussed ridge regression as a modification of linear regression to overcome the empirical non-identifiability of the latter when confronted with high-dimensional data. The means to this end was the addition of a (ridge) penalty to the sum-of-squares loss function of the linear regression model, which

turned out to be equivalent to constraining the parameter domain. This warranted the identification of the regression coefficients, but came at the cost of introducing bias in the estimates. Several properties of ridge regression like moments, MSE, and its Bayesian interpretation have been reviewed. Finally, its behaviour and use have been illustrated in simulation and omics data.

1.13 Exercises

Question 1.1 [†]
Find the ridge regression solution for the data below for a general value of λ and for the straight line model $Y = \beta_0 + \beta_1 X + \varepsilon$ (only apply the ridge penalty to the slope parameter, not to the intercept). Show that when λ is chosen as 0.4, the ridge solution fit is $\hat{Y} = 40 + 1.75X$. Data: $\mathbf{X}^\top = (X_1, X_2, \ldots, X_8)^\top = (-2, -1, -1, -1, 0, 1, 2, 2)^\top$, and $\mathbf{Y}^\top = (Y_1, Y_2, \ldots, Y_8)^\top = (35, 40, 36, 38, 40, 43, 45, 43)^\top$.

Question 1.2 [‡]
Show that the ridge regression estimates can be obtained by ordinary least squares regression on an augmented data set. We augment the centered matrix $\mathbf{X}$ with p additional row $\sqrt{\lambda}\mathbf{I}$, and augment $\mathbf{y}$ with p zeros.

Question 1.3
The coefficients $\boldsymbol{\beta}$ of a linear regression model, $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, are estimated by $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$. The associated fitted values then given by $\widehat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} = \mathbf{HY}$, where $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ referred to as the hat matrix. The matrix $\mathbf{P}$ is a projection matrix and satisfies $\mathbf{H} = \mathbf{H}^2$. Hence, linear regression projects the response $\mathbf{Y}$ onto the vector space spanned by the columns of $\mathbf{Y}$. Consequently, the residuals $\hat{\boldsymbol{\varepsilon}}$ and $\widehat{\mathbf{Y}}$ are orthogonal. Now consider the ridge estimator of the regression coefficients: $\hat{\boldsymbol{\beta}}(\lambda) = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top\mathbf{Y}$. Let $\hat{\mathbf{Y}}(\lambda) = \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)$ be the vector of associated fitted values.
  a) Show that the matrix $\mathbf{H}(\lambda) = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{pp})^{-1}\mathbf{X}^\top$, associated with ridge regression, is not a projection matrix (for any $\lambda > 0$).
  b) Show that the 'ridge fit' $\hat{\mathbf{Y}}(\lambda)$ is not orthogonal to the associated 'ridge residuals' $\hat{\boldsymbol{\varepsilon}}(\lambda)$ (for any $\lambda > 0$).
  c) Derive the distribution of the 'ridge residuals'.

Question 1.4
Recall that there exists $\lambda > 0$ such that $MSE(\hat{\boldsymbol{\beta}}) > MSE[\hat{\boldsymbol{\beta}}(\lambda)]$. Verify that this carries over to the linear predictor. That is, there exists a $\lambda > 0$ such that $MSE(\widehat{\mathbf{Y}}) = MSE(\mathbf{X}\hat{\boldsymbol{\beta}}) > MSE[\mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)]$.

Question 1.5
Consider a 3-gene pathway. Expression levels of these three genes have been measured in an observational study involving hundred individuals. In order to assess how the expression levels of gene A are affect by that of genes B and C, a medical researcher fits the

$$Y_i^{(A)} \quad = \quad \beta_b Y_i^{(B)} + \beta_c Y_i^{(C)} + \varepsilon_i,$$

with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. This model fitted by means of ridge regression, but with a separate penalty parameter, $\lambda_{2,b}$ and $\lambda_{2,c}$, for the two regression coefficient, β_b and β_c, respectively.
  a) Write down the ridge penalized loss function employed by the researcher.
  b) Does a different choice of penalty parameter for the second regression coefficient affect the estimation of the first regression coefficient? Motivate your answer.
  c) The researcher decides that the second covariate $Y_i^{(C)}$ is irrelevant. Instead of removing the covariate from model, the researcher decides to set $\lambda_{2,c} = \infty$. Show that this results in the same ridge estimate for β_b as when fitting (again by means of ridge regression) the model without the second covariate.

Question 1.6
The expression levels of the j-the gene are explained by a linear regression model in terms of those of all

---
[†]This exercise is freely rendered from Draper and Smith (1998)
[‡]This exercise is freely rendered from Hastie *et al.* (2009), but can be found in many other places. The original source is unknown to the author.

other genes. Consider the following two ridge regression estimators of the regression parameter of this model, defined as:

$$\arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} (Y_{i,j} - \mathbf{Y}_{i,\backslash j}\boldsymbol{\beta}_j)^2 + \lambda\|\boldsymbol{\beta}_j\|_2^2 \quad \text{and} \quad \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} (Y_{i,j} - \mathbf{Y}_{i,\backslash j}\boldsymbol{\beta}_j)^2 + n\lambda\|\boldsymbol{\beta}_j\|_2^2.$$

Which do you prefer? Motivate.

2   Generalizing ridge regression

The exposé on ridge regression may be generalized in many ways. Among others different generalized linear models may be considered (confer Section 3). In this section we stick to the linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with the usual assumptions, but fit it in weighted fashion and generalize the common, spherical penalty. The loss function corresponding to this scenario is:

$$(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^{\top}\mathbf{W}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^{\top}\boldsymbol{\Delta}(\boldsymbol{\beta} - \boldsymbol{\beta}_0), \tag{2.1}$$

which comprises a weighted least squares criterion and a generalized ridge penalty. In this $\mathbf{W}$ is a $(n \times n)$-dimensional, diagonal matrix with $(\mathbf{W})_{ii} \in [0, 1]$ representing the weight of the i-th observation. The penalty is now a quadratic form with penalty parameter $\boldsymbol{\Delta}$, a $(p \times p)$-dimensional, positive definite, symmetric matrix. When $\boldsymbol{\Delta} = \lambda\mathbf{I}_{pp}$, one regains the spherical penalty of 'regular ridge regression'. This penalty shrinks each element of the regression parameter $\boldsymbol{\beta}$ equally along the unit vectors $\mathbf{e}_j$. Generalizing $\boldsymbol{\Delta}$ to the class of symmetric, positive definite matrices $\mathcal{S}_{++}$ allows for *i)* different penalization per regression parameter, and *ii)* joint (or correlated) shrinkage among the elements of $\boldsymbol{\beta}$. The penalty parameter $\boldsymbol{\Delta}$ determines the speed and direction of shrinkage. The p-dimensional column vector $\boldsymbol{\beta}_0$ is a user-specified, non-random target towards which $\boldsymbol{\beta}$ is shrunken as the penalty parameter increases. When recasting generalized ridge estimation as a constrained estimation problem, the implications of the penalty may be visualized (Figure 2.1, left panel). The generalized ridge penalty is a quadratic form centered around $\boldsymbol{\beta}_0$. In Figure 2.1 the parameter constraint clearly is ellipsoidal (and not spherical). Moreover, the center of this ellipsoid is not at zero.
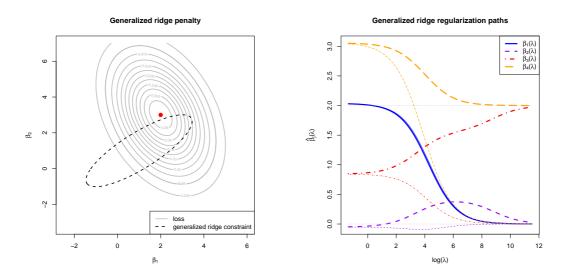

Figure 2.1: Left panel: the contours of the likelihood (grey solid ellipsoids) and the parameter constraint implied by the generalized penalty (black dashed ellipsoid. Right panel: generalized (fat coloured lines) and 'regular' (thin coloured lines) regularization paths of four regression coefficients. The dotted grey (straight) lines indicated the targets towards the generalized ridge penalty shrinks regression coefficient estimates.

The addition of the generalized ridge penalty to the sum-of-squares ensures the existence of a unique regression estimator in the face of super-collinearity. The generalized penalty is a non-degenerated quadratic form in $\boldsymbol{\beta}$ due to the positive definiteness of the matrix $\boldsymbol{\Delta}$. As it is non-degenerate, it is

strictly convex. Consequently, the generalized ridge regression loss function (2.1), being the sum of a convex and strictly convex function, is also strictly convex. This warrants the existence of a unique global minimum and, thereby, a unique estimator.

Like for the 'regular' ridge loss function (1.8), there is an explicit expression for the optimum of the generalized ridge loss function (2.1). To see this, obtain the estimating equation of $\boldsymbol{\beta}$ through equating its derivative with respect to $\boldsymbol{\beta}$ to zero:

$$2\mathbf{X}^\top\mathbf{W}\mathbf{Y} - 2\mathbf{X}^\top\mathbf{W}\mathbf{X}\boldsymbol{\beta} - 2\boldsymbol{\Delta}\boldsymbol{\beta} + 2\boldsymbol{\Delta}\boldsymbol{\beta}_0 \;\;=\;\; \mathbf{0}_p.$$

This is solved by:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta}) \;\;=\;\; (\mathbf{X}^\top\mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}(\mathbf{X}^\top\mathbf{W}\mathbf{Y} + \boldsymbol{\Delta}\boldsymbol{\beta}_0). \tag{2.2}$$

Clearly, this reduces to the 'regular' ridge estimator by setting $\mathbf{W} = \mathbf{I}_{nn}$, $\boldsymbol{\beta}_0 = \mathbf{0}_p$, and $\boldsymbol{\Delta} = \lambda\mathbf{I}_{pp}$. The effects of the generalized ridge penalty on the estimates can be seen in the regularization paths of the estimates. Figure 2.1 (right panel) contains an example of the regularization paths for coefficients of a linear regression model with four explanatory variables. Most striking is the limiting behaviour of the estimates of β_3 and β_4 for large values of the penalty parameter λ: they convergence to a non-zero value (as was specified by a nonzero $\boldsymbol{\beta}_0$). More subtle is the (temporary) convergence of the regularization paths of the estimates of β_2 and β_3. That of β_2 is pulled away from zero (its true value and approximately its unpenalized estimate) towards the estimate of β_3. In the regularization path of β_3 this can be observed in a delayed convergence to its nonzero target value (for comparison consider that of β_4). For reference the corresponding regularization paths of the 'regular' ridge estimates (as thinner lines of the same colour) are included in Figure 2.1.

Example 2.1 *Fused ridge estimation*
An example of a generalized ridge penalty is the *fused ridge penalty*. Consider the standard linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. The fused ridge estimator of $\boldsymbol{\beta}$ then minimizes:

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\sum_{j=2}^{p}\|\beta_j - \beta_{j-1}\|_2^2. \tag{2.3}$$

The penalty in the loss function above can be written as a generalized ridge penalty:

$$\lambda\sum_{j=2}^{p}\|\beta_j - \beta_{j-1}\|_2^2 \;\;=\;\; \boldsymbol{\beta}^\top \begin{pmatrix} \lambda & -\lambda & 0 & \ldots & \ldots & 0 \\ -\lambda & 2\lambda & -\lambda & \ddots & & \vdots \\ 0 & -\lambda & 2\lambda & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -\lambda \\ 0 & \ldots & \ldots & 0 & -\lambda & \lambda \end{pmatrix} \boldsymbol{\beta}.$$
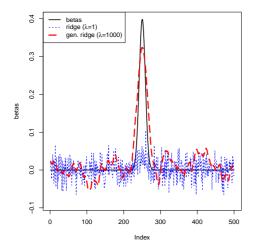
The matrix $\boldsymbol{\Delta}$ employed above is semi-positive definite and therefore the loss function (2.3) is not strictly convex. Hence, often a regular ridge penalty $\|\boldsymbol{\beta}\|_2^2$ is added (with its own penalty parameter).

To illustrate the effect of the fused ridge penalty on the estimation of the linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, let $\beta_j = \phi_{0,1}(z_j)$ with $z_j = -30 + \frac{6}{50}j$ for $j = 1, \ldots, 500$. Sample the elements of the design matrix $\mathbf{X}$ and those of the error vector $\boldsymbol{\varepsilon}$ from the standard normal distribution, then form the response $\mathbf{Y}$ from the linear model. The regression parameter is estimated through fused ridge loss minimization with $\lambda = 1000$. The estimate is shown in Figure 2.2 (red line). For reference the figure includes the true $\boldsymbol{\beta}$ (black line) and the 'regular ridge' estimate with $\lambda = 1$ (blue line). Clearly, the fused ridge estimate yields a nice smooth vector of $\boldsymbol{\beta}$ estimates $\square$

2.1 Moments

The expectation and variance of $\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})$ are obtained through application of the same matrix algebra and expectation and covariance rules used in the derivation of their counterparts of the 'regular' ridge regression estimator. This leads to:

$$\begin{aligned} \mathbb{E}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] &\;\;=\;\; (\mathbf{X}^\top\mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}(\mathbf{X}^\top\mathbf{W}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\Delta}\boldsymbol{\beta}_0), \\ \mathrm{Var}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] &\;\;=\;\; \sigma^2(\mathbf{X}^\top\mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}\mathbf{X}^\top\mathbf{W}^2\mathbf{X}(\mathbf{X}^\top\mathbf{W}\mathbf{X} + \boldsymbol{\Delta})^{-1}. \end{aligned}$$
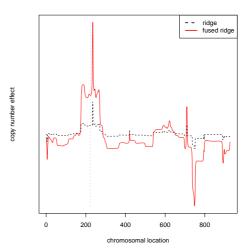
Figure 2.2: Left panel: illustration of the fused ridge estimator (in simulation). The true parameter $\boldsymbol{\beta}$ and its ridge and fused ridge estimates against their spatial order. Right panel: Ridge vs. fused ridge estimates of the DNA copy effect on KRAS expression levels. The dashed, grey vertical bar indicates the location of the KRAS gene.

From these expressions similar limiting behaviour as for the 'regular' ridge regression case can be deduced. To this end let $\mathbf{V}_\delta \mathbf{D}_\delta \mathbf{V}_\delta^\top$ be the eigendecomposition of $\boldsymbol{\Delta}$ and $d_{\delta,j} = (\mathbf{D}_\delta)_{jj}$. Furthermore, define (with some abuse of notation) $\lim_{\boldsymbol{\Delta} \to \infty}$ as the limit of all $d_{\delta,j}$ simultaneously tending to infinity. Then, $\lim_{\boldsymbol{\Delta} \to \infty} \mathbb{E}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] = \boldsymbol{\beta}_0$ and $\lim_{\boldsymbol{\Delta} \to \infty} \mathrm{Var}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] = \mathbf{0}_{pp}$.

Example 2.2
Let $\mathbf{X}$ be an $(n \times p)$-dimensional, orthonormal design matrix. Contrast the regular and generalized ridge regression estimator, the latter with $\mathbf{W} = \mathbf{I}_{pp}$, $\boldsymbol{\beta}_0 = \mathbf{0}_p$ and $\boldsymbol{\Delta} = \lambda \mathbf{R}$ where $\mathbf{R} = (1 - \rho)\mathbf{I}_{pp} + \rho \mathbf{1}_{pp}$ for $\rho \in (-(p-1)^{-1}, 1)$. For $\rho = 0$ the two estimators coincide. The variance of the generalized ridge regression estimator then is $\mathrm{Var}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})] = (\mathbf{I}_{pp} + \boldsymbol{\Delta})^{-2}$. The efficiency of this estimator, measured by its generalized variance, is:

$$\det\{\mathrm{Var}[\hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})]\} \quad = \quad \{[1 + \lambda + (p-1)\rho](1 + \lambda - \rho)^{p-1}\}^{-2}.$$

This efficiency attains its minimum at $\rho = 0$. In the present case, the regular ridge regression estimator is thus more efficient than its generalized counterpart. $\qquad \square$

Example 2.3 *(MSE with perfect target)*
Set $\boldsymbol{\beta}_0 = \boldsymbol{\beta}$, i.e. the target is equal to the true value of the regression parameter. Then:

$$\mathbb{E}[\hat{\boldsymbol{\beta}}(\Delta)] \quad = \quad (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \Delta)^{-1}(\mathbf{X}^\top \mathbf{W} \mathbf{X} \boldsymbol{\beta} + \Delta \boldsymbol{\beta}) \quad = \quad \boldsymbol{\beta}.$$

Hence, irrespective of the choice of Δ, the generalized ridge is then unbiased. Thus:

$$\begin{aligned} \mathrm{MSE}[\hat{\boldsymbol{\beta}}(\Delta)] \quad &= \quad \mathrm{tr}\{\mathrm{Var}[\hat{\boldsymbol{\beta}}(\Delta)]\} \\ &= \quad \mathrm{tr}[\sigma^2 (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \boldsymbol{\Delta})^{-1} \mathbf{X}^\top \mathbf{W}^2 \mathbf{X} (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \boldsymbol{\Delta})^{-1}] \\ &= \quad \sigma^2 \mathrm{tr}[\mathbf{X}^\top \mathbf{W}^2 \mathbf{X} (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \boldsymbol{\Delta})^{-2}]. \end{aligned}$$

When $\boldsymbol{\Delta} = \lambda \mathbf{I}_{pp}$, this MSE is smaller than that of the ML regression estimator, irrespective of the choice of λ. $\qquad \square$

2.2 The Bayesian connection

This generalized ridge estimator can, like the regular ridge estimator, be viewed as a Bayesian estimator. It requires to replace the conjugate prior on $\boldsymbol{\beta}$ by a more general normal law, $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \sigma^2 \boldsymbol{\Delta}^{-1})$, but

retains the gamma prior on σ^2. The joint posterior distribution of $\boldsymbol{\beta}$ and σ^2 is then obtained analogously (the details are left as Exercise 2.2) to Section 1.6:

$$
\begin{aligned}
f_{\boldsymbol{\beta},\sigma^2}(\boldsymbol{\beta}, \sigma^2 \mid \mathbf{Y}, \mathbf{X}) &= f_Y(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\beta}, \sigma^2) \, f_{\boldsymbol{\beta}}(\boldsymbol{\beta}|\sigma^2) \, f_\sigma(\sigma^2) \\
&\propto g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \mid \sigma^2, \mathbf{Y}, \mathbf{X}) \, g_{\sigma^2}(\sigma^2 \mid \mathbf{Y}, \mathbf{X})
\end{aligned}
$$

with

$$
g_{\boldsymbol{\beta}}(\boldsymbol{\beta} \mid \sigma^2, \mathbf{Y}, \mathbf{X}) \quad \propto \quad \exp\left\{ -\frac{1}{2\sigma^2} \left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\boldsymbol{\Delta}) \right]^\top (\mathbf{X}^\top \mathbf{X} + \boldsymbol{\Delta}) \left[\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}(\boldsymbol{\Delta}) \right] \right\}.
$$

This implies $\mathbb{E}(\boldsymbol{\beta} \mid \sigma^2, \mathbf{Y}, \mathbf{X}) = \hat{\boldsymbol{\beta}}(\boldsymbol{\Delta})$. Hence, the generalized ridge regression estimator too can be viewed as the Bayesian posterior mean estimator of $\boldsymbol{\beta}$ when imposing a multivariate Gaussian prior on the regression parameter.

2.3 Application

An illustration involving omics data can be found in the explanation of a gene's expression levels in terms of its DNA copy number. The latter is simply the number of gene copies encoded in the DNA. For instance, for most genes on the autosomal chromosomes the DNA copy number is two, as there is a single gene copy on each chromosome and autosomal chromosomes come in pairs. Alternatively, in males the copy number is one for genes that map to the X or Y chromosome, while in females it is zero for genes on the Y chromosome. In cancer the DNA replication process has often been compromised leading to a (partially) reshuffled and aberrated DNA. Consequently, the cancer cell may exhibit gene copy numbers well over a hundred for classic oncogenes. A faulted replication process does – of course – not nicely follow the boundaries of gene encoding regions. This causes contiguous genes to commonly share aberrated copy numbers. With genes being transcribed from the DNA and a higher DNA copy number implying an enlarged availability of the gene's template, the latter is expected to lead to elevated expression levels. Intuitively, one expects this effect to be localized (a so-called *cis*-effect), but some suggest that aberrations elsewhere in the DNA may directly affect the expression levels of distant genes (referred to as a *trans*-effect).

The *cis*- and *trans*-effects of DNA copy aberrations on the expression levels of the KRAS oncogene in colorectal cancer are investigated. Data of both molecular levels from the TCGA (The Cancer Genome Atlas) repository are downloaded (Cancer Genome Atlas Network, 2012). The gene expression data are limited to that of KRAS, while for the DNA copy number data only that of chromosome 12, which harbors KRAS, is retained. This leaves genomic profiles of 195 samples comprising 927 aberrations. Both molecular data types are zero centered feature-wise. Moreover, the data are limited to ten – conveniently chosen? – samples. The KRAS expression levels are explained by the DNA copy number aberrations through the linear regression model. The model is fitted by means of ridge regression, with $\lambda\boldsymbol{\Delta}$ and $\boldsymbol{\Delta} = \mathbf{I}_{pp}$ and a single-banded $\boldsymbol{\Delta}$ with unit diagonal and the elements of the first off-diagonal equal to the arbitrary value of -0.4. The latter choice appeals to the spatial structure of the genome and encourages similar regression estimates for contiguous DNA copy numbers. The penalty parameter is chosen by means of leave-one-out cross-validation using the squared error loss.

Listing 2.1 R code

```r
# load libraries
library(cgdsr)
library(biomaRt)
library(Matrix)

# get list of human genes
ensembl  <- useMart("ensembl", dataset="hsapiens_gene_ensembl")
geneList <- getBM(attributes=c("ensembl_gene_id", "external_gene_name",
                               "entrezgene_trans_name", "chromosome_name",
                               "start_position", "end_position"), mart=ensembl)

# remove all gene without entrezID
geneList <- geneList[!is.na(geneList[,3]),]
```

```r
# remove all genes not mapping to chr 12
geneList <- geneList[which(geneList[,4] %in% c(12)),]
geneList <- geneList[,-1]
geneList <- unique(geneList)
geneList <- geneList[order(geneList[,3], geneList[,4], geneList[,5]),]

# create CGDS object
mycgds    <- CGDS("http://www.cbioportal.org/public-portal/")

# get available case lists (collection of samples) for a given cancer study
mycancerstudy <- getCancerStudies(mycgds)[37,1]
mycaselist    <- getCaseLists(mycgds,mycancerstudy)[1,1]

# get available genetic profiles
mrnaProf      <- getGeneticProfiles(mycgds,mycancerstudy)[c(4),1]
cnProf        <- getGeneticProfiles(mycgds,mycancerstudy)[c(6),1]

# get data slices for a specified list of genes, genetic profile and case list
cnData   <- numeric()
geData   <- numeric()
geneInfo <- numeric()
for (j in 1:nrow(geneList)){
  geneName <- as.character(geneList[j,1])
  geneData <- getProfileData(mycgds, geneName, c(cnProf, mrnaProf), mycaselist)
  if (dim(geneData)[2] == 2 & dim(geneData)[1] > 0){
    cnData <- cbind(cnData, geneData[,1])
    geData <- cbind(geData, geneData[,2])
    geneInfo <- rbind(geneInfo, geneList[j,])
  }
}
colnames(cnData) <- rownames(geneData)
colnames(geData) <- rownames(geneData)

# preprocess data
Y  <- geData[, match("KRAS", geneInfo[,1]), drop=FALSE]
Y  <- Y - mean(Y)
X  <- sweep(cnData, 2, apply(cnData, 2, mean))

# subset data
idSample <- c(50, 58, 61, 75, 66, 22, 67, 69, 44, 20)
Y        <- Y[idSample]
X        <- X[idSample,]

# generate banded penalty matrix
diags <- list(rep(1, ncol(X)), rep(-0.4, ncol(X)-1))
Delta <- as.matrix(bandSparse(ncol(X), k=-c(0:1), diag=c(diags), symm=TRUE))

# define loss function
CVloss <- function(lambda, X, Y, Delta){
    loss <- 0
    for (i in 1:nrow(X)){
        betasLoo <- solve(crossprod(X[-i,]) + lambda * Delta) %*%
                        crossprod(X[-i,], Y[-i])
        loss <- loss + as.numeric((Y[i] - X[i,,drop=FALSE] %*% betasLoo)^2)
    }
    return(loss)
}

# optimize penalty parameter
limitsL <- c(10^(-10), 10^(10))
optLr   <- optimize(CVloss, limitsL, X=X, Y=Y, Delta=diag(ncol(X)))$minimum
```

```
optLgr <- optimize(CVloss, limitsL, X=X, Y=Y, Delta=Delta)$minimum

# evaluate (generalized) ridge estimators
betasGr <- solve(crossprod(X) + optLgr * Delta)         %*% crossprod(X, Y)
betasR  <- solve(crossprod(X) + optLr  * diag(ncol(X))) %*% crossprod(X, Y)

# plot estimates vs. location
ylims <- c(min(betasR, betasGr), max(betasR, betasGr))
plot(betasR, type="l", ylim=ylims, ylab="copy␣number␣effect",
            lty=2,    yaxt="n",    xlab="chromosomal␣location")
lines(betasGr, lty=1, col="red")
lines(seq(ylims[1], ylims[2], length.out=50) ~
      rep(match("KRAS", geneInfo[,1]), 50), col="grey", lwd=2, lty=3)
legend("topright", c("ridge", "fused␣ridge"), lwd=2,
                  col=c("black", "red"),      lty=c(2, 1))
```

The right panel of Figure 2.2 shows the ridge regression estimate with both choices of $\mathbf{\Delta}$ and optimal penalty parameters plotted against the chromosomal order. The location of KRAS is indicated by a vertical dashed bar. The ordinary ridge regression estimates show a minor peak at the location of KRAS but is otherwise more or less flat. In the generalized ridge estimates the peak at KRAS is emphasized. Moreover, the region close to KRAS exhibits clearly elevated estimates, suggesting co-abberated DNA. For the remainder the generalized ridge estimates portray a flat surface, with the exception of a single downward spike away from KRAS. Such negative effects are biologically nonsensible (more gene templates leading to reduced expression levels?). On the whole the generalized ridge estimates point towards the *cis*-effect as the dominant genomic regulation mechanism of KRAS expression. The isolated spike may suggest the presence of a *trans*-effect, but its sign is biological nonsensible and the spike is fully absent in the ordinary ridge estimates. This leads us to ignore the possibility of a genomic *trans*-effect on KRAS expression levels in colorectal cancer.

The sample selection demands justification. It yields a clear illustrate-able difference between the ordinary and ridge estimates. When all samples are left in, the *cis*-effect is clearly present, discernable from both estimates that yield a virtually similar profile.

2.4 Generalized ridge regression

What is generally referred to as 'generalized ridge regression' (cf. Hoerl and Kennard, 1970; Hemmerle, 1975) is the particular case of loss function (2.1) in which $\mathbf{W} = \mathbf{I}_{nn}$, $\boldsymbol{\beta}_0 = \mathbf{0}_p$, and $\mathbf{\Delta} = \mathbf{V}_x \mathbf{\Lambda} \mathbf{V}_x^\top$, where $\mathbf{V}_x$ is obtained from the singular value decomposition of $\mathbf{X}$ (i.e., $\mathbf{X} = \mathbf{U}_x \mathbf{D}_x \mathbf{V}_x^\top$ with its constituents endowed with the usual interpretation) and $\mathbf{\Lambda}$ a positive definite diagonal matrix. This gives the estimator:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}(\mathbf{\Lambda}) &= (\mathbf{X}^\top \mathbf{X} + \mathbf{\Delta})^{-1} \mathbf{X}^\top \mathbf{Y} \\
&= (\mathbf{V}_x \mathbf{D}_x \mathbf{U}_x^\top \mathbf{U}_x \mathbf{D}_x \mathbf{V}_x^\top + \mathbf{V}_x \mathbf{\Lambda} \mathbf{V}_x^\top)^{-1} \mathbf{V}_x \mathbf{D}_x \mathbf{U}_x \mathbf{Y} \\
&= \mathbf{V}_x (\mathbf{D}_x^2 + \mathbf{\Lambda})^{-1} \mathbf{D}_x \mathbf{U}_x \mathbf{Y}.
\end{aligned}
$$

From this last expression it becomes clear how this estimator generalizes the 'regular ridge estimator'. The latter shrinks all eigenvalues, irrespectively of their size, in the same manner through a common penalty parameter. The 'generalized ridge estimator', through differing penalty parameters (i.e. the diagonal elements of $\mathbf{\Lambda}$), shrinks them individually.

The generalized ridge estimator coincides with the Bayesian linear regression estimator with the normal prior $\mathcal{N}[\mathbf{0}_p, (\mathbf{V}_x \mathbf{\Lambda} \mathbf{V}_x^\top)^{-1}]$ on the regression parameter $\boldsymbol{\beta}$ (and preserving the inverse gamma prior on the error variance). Assume $\mathbf{X}$ to be of full column rank and choose $\mathbf{\Lambda} = g^{-1} \mathbf{D}_x^2$ with g a positive scalar. The prior on $\boldsymbol{\beta}$ then – assuming $(\mathbf{X}^\top \mathbf{X})^{-1}$ exits – reduces to Zellner's g-prior: $\boldsymbol{\beta} \sim \mathcal{N}[\mathbf{0}_p, g(\mathbf{X}^\top \mathbf{X})^{-1}]$ (Zellner, 1986). The corresponding estimator of the regression coefficient is: $\hat{\boldsymbol{\beta}}(g) = g(1+g)^{-1}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$, which is proportional to the unpenalized ordinary least squares estimator of $\boldsymbol{\beta}$.

For convenience of notation in the analysis of the generalized ridge estimator the linear regression model is usually rewritten as:

$$
\mathbf{Y} \;=\; \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \;=\; \mathbf{X}\mathbf{V}_x \mathbf{V}_x^\top \boldsymbol{\beta} + \boldsymbol{\varepsilon} \;=\; \tilde{\mathbf{X}}\boldsymbol{\alpha} + \boldsymbol{\varepsilon},
$$

with $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{V}_x = \mathbf{U}_x\mathbf{D}_x$ (and thus $\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}} = \mathbf{D}_x^2$) and $\boldsymbol{\alpha} = \mathbf{V}_x^\top\boldsymbol{\beta}$ with loss function $(\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\alpha})^\top(\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\alpha}) + \boldsymbol{\alpha}^\top\boldsymbol{\Lambda}\boldsymbol{\alpha}$. In the notation above the generalized ridge estimator is then:

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda}) \quad = \quad (\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}} + \boldsymbol{\Lambda})^{-1}\tilde{\mathbf{X}}^\top\mathbf{Y} = (\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}\tilde{\mathbf{X}}^\top\mathbf{Y},$$

from which one obtains $\hat{\boldsymbol{\beta}}(\boldsymbol{\Lambda}) = \mathbf{V}_x\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})$. Using $\mathbb{E}[\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})] = (\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}\mathbf{D}_x^2\boldsymbol{\alpha}$ and $\mathrm{Var}[\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})] = \sigma^2(\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}\mathbf{D}_x^2(\mathbf{D}_x^2 + \boldsymbol{\Lambda})^{-1}$, the MSE for the generalized ridge estimator can be written as:

$$\mathrm{MSE}[\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})] \quad = \quad \sum_{j=1}^{p}(\sigma^2 d_{x,j}^2 + \alpha_j^2\lambda_j^2)(d_{x,j}^2 + \lambda_j)^{-2},$$

where $d_{x,j} = (\mathbf{D}_x)_{jj}$ and $\lambda_j = (\boldsymbol{\Lambda})_{jj}$. Having $\boldsymbol{\alpha}$ and σ^2 available, it is easily seen (equate the derivative w.r.t. λ_j to zero and solve) that the MSE of $\hat{\boldsymbol{\alpha}}(\boldsymbol{\Lambda})$ is minimized by $\lambda_j = \sigma^2/\alpha_j^2$ for all j. With $\boldsymbol{\alpha}$ and σ^2 unknown, Hoerl and Kennard (1970) suggest an iterative procedure to estimate the λ_j's. Initiate the procedure with the OLS estimates of $\boldsymbol{\alpha}$ and σ^2, followed by sequentially updating the λ_j's and the estimates of $\boldsymbol{\alpha}$ and σ^2. An analytic expression of the limit of this procedure exists (Hemmerle, 1975). This limit, however, still depends on the observed $\mathbf{Y}$ and as such it does not necessarily yield the minimal attainable value of the MSE. This limit may nonetheless still yield a potential gain in MSE. This is investigated in Lawless (1981). Under a variety of cases it seems to indeed outperform the OLS estimator, but there are exceptions.

2.5 Conclusion

To conclude: a note of caution. The generalized ridge penalty is extremely flexible. It can incorporate any prior knowledge on the parameter values (through specification of $\boldsymbol{\beta}_0$) and the relations among these parameters (via $\boldsymbol{\Delta}$). While a pilot study or literature may provide a suggestion for $\boldsymbol{\beta}_0$, it is less obvious how to choose an informative $\boldsymbol{\Delta}$ (although a spatial structure is a nice exception). In general, exact knowledge on the parameters should not be incorporated implicitly via the penalty (read: prior) but preferably be used explicitly in the model – the likelihood – itself. Though this may be the viewpoint of a prudent frequentist and a subjective Bayesian might disagree.

2.6 Exercises

Question 2.1
Consider the linear regression model $Y_i = \beta_1 X_{i,1} + \beta_2 X_{i,2} + \varepsilon_i$ for $i = 1, \ldots, n$. Suppose estimates of the regression parameters (β_1, β_2) of this model are obtained through the minimization of the sum-of-squares augmented with a ridge-type penalty:

$$\left[\sum_{i=1}^{n}(Y_i - \beta_1 X_{i,1} - \beta_2 X_{i,2})^2 \right] + \lambda(\beta_1^2 + \beta_2^2 + 2\nu\beta_1\beta_2),$$

with penalty parameters $\lambda \in \mathbb{R}_{>0}$ and $\nu \in (-1, 1)$.

    a) Sketch (for both $\nu = 0$ and $\nu = 0.9$) the shape of the parameter constraint induced by the penalty above and describe in words the qualitative difference between both shapes.

    b) When $\nu = -1$ and $\lambda \to \infty$ the estimates of β_1 and β_2 (resulting from minimization of the penalized loss function above) converge towards each other: $\lim_{\lambda\to\infty}\hat{\beta}_1(\lambda, -1) = \lim_{\lambda\to\infty}\hat{\beta}_2(\lambda, -1)$. Motivated by this observation a data scientists incorporates the equality constraint $\beta_1 = \beta = \beta_2$ explicitly into the model, and s/he estimates the 'joint regression parameter' β through the minimization (with respect to β) of:

$$\left[\sum_{i=1}^{n}(Y_i - \beta X_{i,1} - \beta X_{i,2})^2 \right] + \delta\beta^2,$$

    with penalty parameter $\delta \in \mathbb{R}_{>0}$. The data scientist is surprised to find that resulting estimate $\hat{\beta}(\delta)$ does not have the same limiting (in the penalty parameter) behavior as the $\hat{\beta}_1(\lambda, -1)$, i.e. $\lim_{\delta\to\infty}\hat{\beta}(\delta) \neq \lim_{\lambda\to\infty}\hat{\beta}_1(\lambda, -1)$. Explain the misconception of the data scientist.

c) Assume that *i)* $n \gg 2$, *ii)* the unpenalized estimates $(\hat{\beta}_1(0,0), \hat{\beta}_2(0,0))$ equal $(-2, 2)$, and *iii)* that the two covariates X_1 and X_2 are zero-centered, have equal variance, and are strongly negatively correlated. Consider $(\hat{\beta}_1(\lambda, \nu), \hat{\beta}_2(\lambda, \nu))$ for both $\nu = -0.9$ and $\nu = 0.9$. For which value of ν do you expect the sum of the absolute value of the estimates to be largest? *Hint:* Distinguish between small and large values of λ and think geometrically!

Question 2.2

Consider the linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_p, \sigma^2 \mathbf{I}_{pp})$. Assume $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\beta}_0, \sigma^2 \boldsymbol{\Delta}^{-1})$ with $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ and $\boldsymbol{\Delta} \succ 0$ and a gamma prior on the error variance. Verify (i.e., work out the details of the derivation) that the posterior mean coincides with the generalized ridge estimator defined as:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X} + \boldsymbol{\Delta})^{-1}(\mathbf{X}^\top \mathbf{Y} + \boldsymbol{\Delta}\boldsymbol{\beta}_0).$$

Question 2.3

The ridge penalty may be interpreted as a multivariate normal prior on the regression coefficients: $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}_{pp})$. Different priors may be considered. In case the covariates are spatially related in some sense (e.g. genomically), it may of interest to assume a first-order autoregressive prior: $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \lambda^{-1}\boldsymbol{\Sigma}_A)$, in which $\boldsymbol{\Sigma}_A$ is a $p \times p$-correlation matrix with $(\boldsymbol{\Sigma}_A)_{j_1, j_2} = \rho^{|j_1 - j_2|}$ for some correlation coefficient $\rho \in [0, 1)$. Hence,

$$\boldsymbol{\Sigma}_A = \begin{pmatrix} 1 & \rho & \cdots & \rho^{p-1} \\ \rho & 1 & \cdots & \rho^{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{p-1} & \rho^{p-2} & \cdots & 1 \end{pmatrix}.$$

a) The penalized loss function associated with this AR(1) prior is:

$$\mathcal{L}(\boldsymbol{\beta}; \lambda, \boldsymbol{\Sigma}_A) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \boldsymbol{\beta}^\top \boldsymbol{\Sigma}_A^{-1} \boldsymbol{\beta}.$$

Find the minimizer of this loss function.

b) What is the effect of ρ on the ridge estimates? Contrast this to the effect of λ. Illustrate this on (simulated) data.

c) Instead of an AR(1) prior assume a prior with a uniform correlation between the elements of $\boldsymbol{\beta}$. That is, replace $\boldsymbol{\Sigma}_A$ by $\boldsymbol{\Sigma}_U$, given by:

$$\boldsymbol{\Sigma}_U = \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix}$$

Investigate (again on data) the effect of changing from the AR(1) to the uniform prior on the ridge regression estimates.

3 Ridge logistic regression

Ridge penalized estimation is not limited to the standard linear regression model, but may be used to estimate (virtually) any model. Here we illustrate how it may be used to fit the logistic regression model. To this end we first recap this model and the (unpenalized) maximum likelihood estimation of its parameters. After which the model is estimated by means of ridge penalized maximum likelihood, which will turn out to be a relatively straightforward modification of unpenalized estimation.

3.1 Logistic regression

The logistic regression model explains a binary response variable (through some transformation) by a linear combination of a set of covariates (as in the linear regression model). Denote this response of the i-th sample by Y_i with $Y_i \in \{0, 1\}$ for $i = 1, \ldots, n$. The n-dimensional column vector $\mathbf{Y}$ stacks these n responses. For each sample information on the p explanatory variables $X_{i,1}, \ldots, X_{i,p}$ is available. In row vector form this information is denoted $\mathbf{X}_{i,*} = (X_{i,1}, \ldots, X_{i,p})$. Or, in short, $\mathbf{X}_i$ when the context tolerates no confusion. The $(n \times p)$-dimensional matrix $\mathbf{X}$ aggregates these vectors, such that $\mathbf{X}_i$ is the i-th row vector.

The binary response cannot be modelled as in the linear model like $Y_i = \mathbf{X}_i \boldsymbol{\beta} + \varepsilon_i$. With each element of $\mathbf{X}_i$ and $\boldsymbol{\beta}$ assuming a value in $\mathbb{R}$, the linear predictor is not restricted to the domain of the response. This is resolved by modeling $p_i = P(Y_i = 1)$ instead. Still the linear predictor may exceed the domain of the response ($p_i \in [0, 1]$). Hence, a transformation is applied to map p_i to $\mathbb{R}$, the range of the linear predictor. The transformation associated with the logistic regression model is the logarithm of the odds, with the odds defined as: $odds = P(\text{succes})/P(\text{failure}) = p_i/(1 - p_i)$. The logistic model is then written as $\log[p_i/(1 - p_i)] = \mathbf{X}_i \boldsymbol{\beta}$ for all i. Or, expressed in terms of the response:

$$p_i \quad = \quad P(Y_i = 1) \quad = \quad g^{-1}(\mathbf{X}_i; \boldsymbol{\beta}) \quad = \quad \frac{\exp(\mathbf{X}_i \boldsymbol{\beta})}{1 + \exp(\mathbf{X}_i \boldsymbol{\beta})}.$$

The function $g(\cdot; \cdot)$ is called the *link function*. It links the response to the explanatory variables. The one above is called the logistic link function. Or short, logit. The regression parameters have tangible interpretations. When the first covariate represents the intercept, i.e. $X_{i,j} = 1$ for all i, then β_1 determines where the link function equals a half when all other covariates fail to contribute to the linear predictor (i.e. where $P(Y_i = 1 \mid \mathbf{X}_i) = 0.5$ when $\mathbf{X}_i \boldsymbol{\beta} = \beta_1$). This is illustrated in the top-left panel of Figure 3.1 for various choices of the intercept. On the other hand, the regression parameters are directly related to the odds ratio: *odds ratio* $= \text{odds}(X_{i,j} + 1)/\text{odds}(X_{i,j}) = \exp(\beta_j)$. Hence, the effect of a unit change in the j-th covariate on the odds ratio is $\exp(\beta_j)$ (see Figure 3.1, top-right panel). Other link functions (depicted in Figure 3.1, bottom-left panel) are common, e.g. the *probit*: $p_i = \Phi_{0,1}(\mathbf{X}_i \boldsymbol{\beta})$; the *cloglog*: $p_i = \frac{1}{\pi} \arctan(\mathbf{X}_i \boldsymbol{\beta}) + \frac{1}{2}$; the *Cauchit*: $p_i = \exp[-\exp(\mathbf{X}_i \boldsymbol{\beta})]$. All these link function are invertible. Irrespective of the choice of the link function, the binary data are thus modelled as $Y_i \sim \mathcal{B}[g^{-1}(\mathbf{X}_i; \boldsymbol{\beta}), 1]$. That is, as a single draw from the Binomial distribution with success probability $g^{-1}(\mathbf{X}_i; \boldsymbol{\beta})$.

Let us now estimate the parameter of the logistic regression model by means of the maximum likelihood method. The likelihood of the experiment is then:

$$L(\mathbf{Y} \mid \mathbf{X}; \boldsymbol{\beta}) \quad = \quad \prod_{i=1}^{n} \big[P(Y_i = 1 \mid \mathbf{X}_i)\big]^{Y_i} \big[P(Y_i = 0 \mid \mathbf{X}_i)\big]^{1-Y_i}.$$
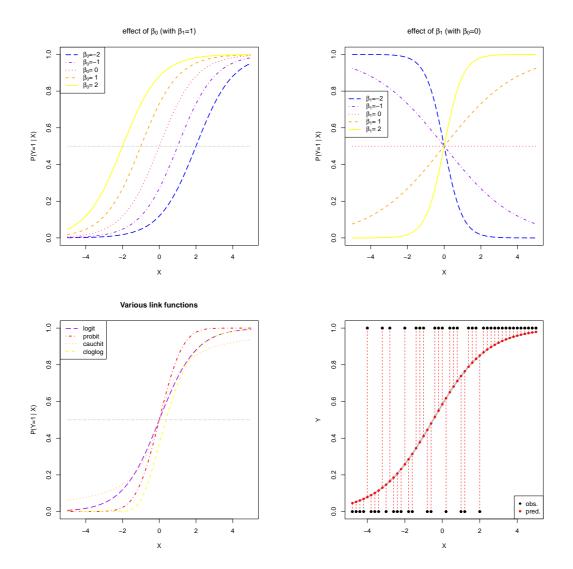
Figure 3.1: Top row, left panel: the response curve for various choices of the intercept β_0. Top row, right panel: the response curve for various choices of the regression coefficent β_1. Bottom row, left panel: the responce curve for various choices of the link function. Bottom panel, right panel: observations, fits and their deviations.

After taking the logarithm and some ready algebra, the log-likelihood is found to be:

$$\mathcal{L}(\mathbf{Y} \mid \mathbf{X}; \boldsymbol{\beta}) \quad = \quad \sum_{i=1}^{n} \left\{ Y_i \mathbf{X}_i \boldsymbol{\beta} - \log[1 + \exp(\mathbf{X}_i \boldsymbol{\beta})] \right\}.$$

Differentiate the log-likelihood with respect to $\boldsymbol{\beta}$, equate it zero, and obtain the estimating equation for $\boldsymbol{\beta}$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} \quad = \quad \sum_{i=1}^{n} \left[Y_i - \frac{\exp(\mathbf{X}_i \boldsymbol{\beta})}{1 + \exp(\mathbf{X}_i \boldsymbol{\beta})} \right] \mathbf{X}_i^{\top} \quad = \quad \mathbf{0}_p. \tag{3.1}$$

The ML estimate of $\boldsymbol{\beta}$ strikes a (weighted by the $\mathbf{X}_i$) balance between observation and model. Put differently (and illustrated in the bottom-right panel of Figure 3.1), a curve is fit through data by minimizing the distance between them: at the ML estimate of $\boldsymbol{\beta}$ a weighted average of their deviations is zero.

The maximum likelihood estimate of $\boldsymbol{\beta}$ is evaluated by solving Equation (3.1) with respect to $\boldsymbol{\beta}$ by means of the Newton-Raphson algorithm. The Newton-Raphson algorithm iteratively finds the zeros

of a smooth enough function $f(\cdot)$. Let x_0 denote an initial guess of the zero. Then, approximate $f(\cdot)$ around x_0 by means of a first order Taylor series: $f(x) \approx x_0 + (x - x_0) \, (df/dx)|_{x=x_0}$. Solve this for x and obtain: $x = x_0 - [(df/dx)|_{x=x_0}]^{-1} f(x_0)$. Let x_1 be the solution for x, use this as the new guess and repeat the above until convergence. When the function $f(\cdot)$ has multiple arguments, is vector-valued and denoted by $\vec{f}$, and the Taylor approximation becomes: $\vec{f}(\mathbf{x}) \approx \mathbf{x}_0 + J\vec{f}\big|_{\mathbf{x}=\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0)$ with

$$
J\vec{f} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_p} \\ \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_q}{\partial x_1} & \frac{\partial f_q}{\partial x_2} & \cdots & \frac{\partial f_q}{\partial x_p} \end{pmatrix},
$$

the Jacobi matrix. An update of x_0 is now readily constructed by solving (the approximation for) $\vec{f}(\mathbf{x}) = \mathbf{0}$ for $\mathbf{x}$.

When applied here to the maximum likelihood estimation of the regression parameter $\boldsymbol{\beta}$ of the logistic regression model, the Newton-Raphson update is:

$$
\hat{\boldsymbol{\beta}}^{\text{new}} = \hat{\boldsymbol{\beta}}^{\text{old}} - \Big(\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \Big)^{-1} \Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{\text{old}}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} \Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^{\text{old}}}
$$

where the Hessian of the log-likelihood equals:

$$
\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = -\sum_{i=1}^n \frac{\exp(\mathbf{X}_i \boldsymbol{\beta})}{[1 + \exp(\mathbf{X}_i \boldsymbol{\beta})]^2} \mathbf{X}_i^\top \mathbf{X}_i.
$$

Iterative application of this updating formula converges to the ML estimate of $\boldsymbol{\beta}$.

The Newton-Raphson algorithm is often reformulated as an iteratively re-weighted least squares algorithm. Hereto, first write the gradient and Hessian in matrix notation:

$$
\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} = \mathbf{X}^\top [\mathbf{Y} - \vec{g}^{-1}(\mathbf{X}; \boldsymbol{\beta})] \quad \text{and} \quad \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = -\mathbf{X}^\top \mathbf{W} \mathbf{X},
$$

where $\vec{g}^{-1}(\mathbf{X}; \boldsymbol{\beta}) = [g^{-1}(\mathbf{X}_{1,*}; \boldsymbol{\beta}), \ldots, g^{-1}(\mathbf{X}_{n,*}; \boldsymbol{\beta})]^\top$ with $g^{-1}(\cdot; \cdot) = \exp(\cdot; \cdot)/[1 + \exp(\cdot; \cdot)]$ and $\mathbf{W}$ diagonal with $(\mathbf{W})_{ii} = \exp(\mathbf{X}_i \hat{\boldsymbol{\beta}}^{\text{old}})[1 + \exp(\mathbf{X}_i \hat{\boldsymbol{\beta}}^{\text{old}})]^{-2}$. The updating formula of the estimate then becomes:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}^{\text{new}} &= \hat{\boldsymbol{\beta}}^{\text{old}} + (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top [\mathbf{Y} - \vec{g}^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \\
&= (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \{ \mathbf{X} \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1} [\mathbf{Y} - \vec{g}^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \} \\
&= (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{Z},
\end{aligned}
$$

where $\mathbf{Z} = \{ \mathbf{X} \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1} [\mathbf{Y} - \vec{g}^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \}$. The Newton-Raphson update is thus the solution to the following weighted least squares problem:

$$
\hat{\boldsymbol{\beta}}^{\text{new}} = \arg\min_{\boldsymbol{\beta}} (\mathbf{Z} - \mathbf{X} \boldsymbol{\beta})^\top \mathbf{W} (\mathbf{Z} - \mathbf{X} \boldsymbol{\beta}).
$$

Effectively, at each iteration the *adjusted response* $\mathbf{Z}$ is regressed on the covariates that comprise $\mathbf{X}$. For more on logistic regression confer the monograph of Hosmer Jr *et al.* (2013).

3.2 Ridge estimation

High-dimensionally, the linear predictor $\mathbf{X} \boldsymbol{\beta}$ may be uniquely defined, but the maximum likelihood estimate of the logistic regression parameter is not. Assume $p > n$ and an estimate $\hat{\boldsymbol{\beta}}$ available. Due to the high-dimensionality, the null space of $\mathbf{X}$ is non-trivial. Hence, let $\boldsymbol{\gamma} \in \text{null}(\text{span}(\mathbf{X}))$. Then: $\mathbf{X} \hat{\boldsymbol{\beta}} = \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{X} \boldsymbol{\gamma} = \mathbf{X}(\hat{\boldsymbol{\beta}} + \boldsymbol{\gamma})$. As the null space is a $p - n$-dimensional subspace, $\boldsymbol{\gamma}$ need not equal zero. Hence, an infinite number of estimates of the logistic regression parameter exists that yield the same log-likelihood. Augmentation of the loss function with a ridge penalty resolves the matter, as their sum
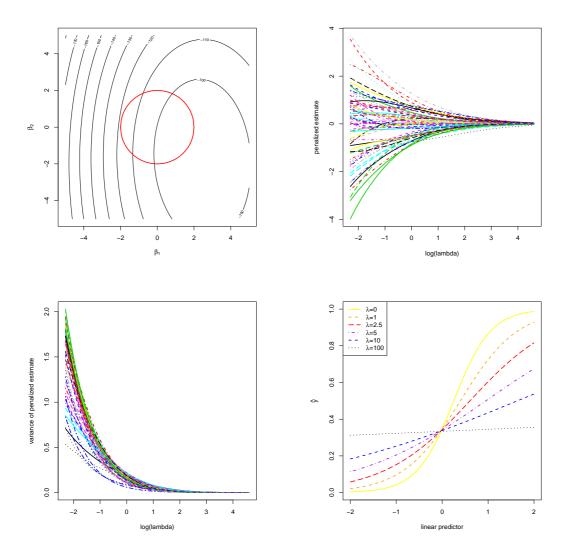
Figure 3.2: Top row, left panel: contour plot of the penalized log-likelihood of a logistic regression model with the ridge constraint (red line). Top row, right panel: the regularization paths of the ridge estimator of the logistic regression parameter. Bottom row, left panel: variance of the ridge estimator of the logistic regression parameter against the logarithm of the penalty parameter. Bottom panel, right panel: the predicted success probability versus the linear predictor for various choices of the penalty parameter.

is strictly concave in $\boldsymbol{\beta}$ (not convex as a maximum rather than a minimum is sought here) and thereby has a unique maximum.

Ridge maximum likelihood estimates of the logistic model parameters are found by the maximization of the ridge penalized loglikelihood (cf. Schaefer *et al.* 1984; Le Cessie and Van Houwelingen 1992):

$$
\begin{aligned}
\mathcal{L}^{\mathrm{pen}}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}, \lambda) &= \mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) - \tfrac{1}{2}\lambda\|\boldsymbol{\beta}\|_2^2 \\
&= \sum_{i=1}^{n}\left\{Y_i\mathbf{X}_i\boldsymbol{\beta} - \log[1 + \exp(\mathbf{X}_i\boldsymbol{\beta})]\right\} - \tfrac{1}{2}\lambda\boldsymbol{\beta}^{\top}\boldsymbol{\beta},
\end{aligned}
$$

where the second summand is the ridge penalty (the sum of the square of the elements of $\boldsymbol{\beta}$) with λ the penalty parameter. Note that as in Section 1.5 maximization of this penalized loss function can be reformulated as a constrained estimation problem. This is illustrated by the top left panel of Figure 3.2, which depicts the contours (black lines) of the log-likelihood and the spherical domain of the parameter (red line). The optimization of the above loss function proceeds, due to the differentiability

of the penalty, fully analogous to the unpenalized case and uses the Newton-Raphson algorithm for solving the (penalized) estimating equation. Hence, the unpenalized ML estimation procedure is modified straightforwardly by replacing gradient and Hessian by their 'penalized' counterparts:

$$\frac{\partial \mathcal{L}^{\text{pen}}}{\partial \boldsymbol{\beta}} \;=\; \frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}} - \lambda \boldsymbol{\beta} \quad \text{and} \quad \frac{\partial^2 \mathcal{L}^{\text{pen}}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \;=\; \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} - \lambda \mathbf{I}_{pp}.$$

With these at hand, the Newton-Raphson algorithm is (again) reformulated as an iteratively re-weighted least squares algorithm with the updating step changes accordingly to:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{\text{new}} &= \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{V}^{-1} \{ \mathbf{X}^\top [\mathbf{Y} - \vec{\mathbf{g}}^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] - \lambda \boldsymbol{\beta}^{\text{old}} \} \\
&= \mathbf{V}^{-1} \mathbf{V} \hat{\boldsymbol{\beta}}^{\text{old}} - \lambda \mathbf{V}^{-1} \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{V}^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{W}^{-1} [\mathbf{Y} - \vec{\mathbf{g}}^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \\
&= \mathbf{V}^{-1} \mathbf{X}^\top \mathbf{W} \{ \mathbf{X} \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1} [\mathbf{Y} - \vec{\mathbf{g}}^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \} \\
&= [\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{Z},
\end{aligned}$$

where $\mathbf{V} = \mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp}$ and $\mathbf{W}$ and $\mathbf{Z}$ as before. Hence, use this to update the estimate of $\boldsymbol{\beta}$ until convergence, which yields the desired ridge ML estimate.

Obviously, the ridge estimate of the logistic regression parameter tends to zero as $\lambda \to \infty$. Now consider a linear predictor with an intercept that is left unpenalized. When λ tends to infinity, all regression coefficients but the intercept vanish. The intercept is left to model the success probability. Hence, in this case $\lim_{\lambda \to \infty} \hat{\beta}_0(\lambda) = \log[\frac{1}{n} \sum_{i=1}^n Y_i / \frac{1}{n} \sum_{i=1}^n (1 - Y_i)]$.

The effect of the ridge penalty on parameter estimates propagates to the predictor $\hat{p}_i$. The linear predictor of the linear regression model involving the ridge estimator $\mathbf{X}_i \hat{\boldsymbol{\beta}}(\lambda)$ shrinks towards a common value for each i, leading to a scale difference between observation and predictor (as seen before in Section 1.11). This behaviour transfers to the ridge logistic regression predictor, as is illustrated on simulated data. The dimension and sample size of these data are $p = 2$ and $n = 200$, respectively. The covariate data are drawn from the standard normal, while that of the response is sampled from a Bernoulli distribution with success probability $P(Y_i = 1) = \exp(2X_{i,1} - 2X_{i,2}) / [1 + \exp(2X_{i,1} - 2X_{i,2})]$. The logistic regression model is estimated from these data by means of ridge penalized likelihood maximization with various choices of the penalty parameter. The bottom right plot in Figure 3.2 shows the predicted success probability versus the linear predictor for various choices of the penalty parameter. Larger values of the penalty parameter λ flatten the slope of this curve. Consequently, for larger λ more excessive values of the covariates are needed to achieve the same predicted success probability as those obtained with smaller λ at more moderate covariate values. The implications for the resulting classification may become clearer when studying the effect of the penalty parameter on the 'failure' and 'success regions' respectively defined by:

$\{(x_1, x_2) : P(\textcolor{green}{\mathbf{Y = 0}} \,|\, X_1 = x_1, X_2 = x_2, \hat{\boldsymbol{\beta}}(\lambda)) > 0.75\}$,
$\{(x_1, x_2) : P(\textcolor{red}{\mathbf{Y = 1}} \,|\, X_1 = x_1, X_2 = x_2, \hat{\boldsymbol{\beta}}(\lambda)) > 0.75\}$.

This separates the design space in a light red ('failure') and light green ('success') domain. The white bar between them is the domain where samples cannot be classified with high enough certainty. As λ grows, so does the white area that separates the failure and success regions. Hence, as stronger penalization shrinks the logistic regression parameter estimate towards zero, it produces a predictor that is less outspoken in its class assignments.

3.3 Moments

The 1st and 2nd order moment of the ridge ML parameter of the logistic model may be approximated by the final update of the Newton-Raphson estimate. Assume the one-to-last update $\hat{\boldsymbol{\beta}}^{\text{old}}$ to be non-random and proceed as for the ridge estimator of the linear regression model parameter to arrive at:

$$\begin{aligned}
\mathbb{E}(\hat{\boldsymbol{\beta}}^{\text{new}}) &= [\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1} \mathbf{X}^\top \mathbf{W} \mathbb{E}(\mathbf{Z}), \\
\text{Var}(\hat{\boldsymbol{\beta}}^{\text{new}}) &= [\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1} \mathbf{X}^\top \mathbf{W} [\text{Var}(\mathbf{Z})] \mathbf{W} \mathbf{X} [\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp}]^{-1},
\end{aligned}$$

with

$$\begin{aligned}
\mathbb{E}(\mathbf{Z}) &= \{ \mathbf{X} \hat{\boldsymbol{\beta}}^{\text{old}} + \mathbf{W}^{-1} [\mathbb{E}(\mathbf{Y}) - \vec{\mathbf{g}}^{-1}(\mathbf{X}; \boldsymbol{\beta}^{\text{old}})] \}, \\
\text{Var}(\mathbf{Z}) &= \mathbf{W}^{-1} \text{Var}(\mathbf{Y}) \mathbf{W}^{-1} = \mathbf{W}^{-1},
\end{aligned}$$

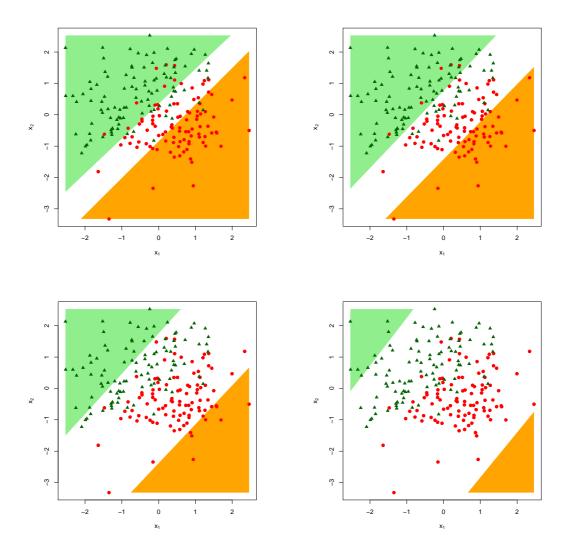**Figure 3.3**: The realized design as scatter plot (X_1 vs X_2 overlayed by the success (RED) and failure regions (GREEN) for various choices of the penalty parameter: $\lambda = 0$ (top row, left panel), $\lambda = 10$ (top row, right panel) $\lambda = 40$ (bottom row, left panel), $\lambda = 100$ (bottom row, right panel).

where the identity $\mathrm{Var}(\mathbf{Y}) = \mathbf{W}$ follows from the variance of a Binomial distributed random variable. From these expressions similar properties as for the ridge ML estimate of the regression parameter of the linear model may be deduced. For instance, the ridge ML estimate of the logistic regression parameter converges to zero as the penalty parameter tends to infinity (confer the top right panel of Figure 3.2). Similarly, their variances vanish as $\lambda \to \infty$ (illustrated in the bottom left panel of Figure 3.2).

3.4 The Bayesian connection

All penalized estimators can be formulated as Bayesian estimators, including the ridge logistic estimator. In particular, ridge estimators correspond to Bayesian estimators with a multivariate normal prior on the regression coefficients. Thus, assume $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}_p, \boldsymbol{\Delta}^{-1})$. The posterior distribution of $\boldsymbol{\beta}$ then is:

$$f_{\boldsymbol{\beta}}(\boldsymbol{\beta} \mid \mathbf{Y}, \mathbf{X}) \quad \propto \quad \left\{ \prod_{i=1}^{n} \left[P(Y_i = 1 \mid \mathbf{X}_i) \right]^{Y_i} \left[P(Y_i = 0 \mid \mathbf{X}_i) \right]^{1-Y_i} \right\} \exp(-\tfrac{1}{2}\boldsymbol{\beta}\boldsymbol{\Delta}\boldsymbol{\beta}).$$

This does not coincide with any standard distribution. But, under appropriate conditions, the posterior distribution is asymptotically normal. This invites a (multivariate) normal approximation to the posterior

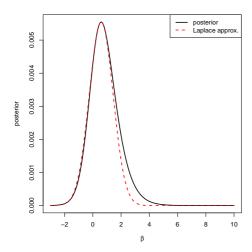distribution above. The Laplace's method provides (cf. Bishop, 2006).

Figure 3.4: Right panel: Laplace approximation to the posterior density of the Bayesian logistic regression parameter.

Laplace's method *i)* centers the normal approximation at the mode of the posterior, and *ii)* chooses the covariance to match the curvature of the posterior at the mode. The posterior mode is the location of the maximum of the posterior distribution. The location of this maximum coincides with that of the logarithm of the posterior. The latter is the log-likelihood augmented with a ridge penalty. Hence, the posterior mode, which is taken as the mean of the approximating Gaussian, coincides with the ridge logistic estimator. For the covariance of the approximating Gaussian, the logarithm of the posterior is approximated by a second order Taylor series around the posterior mode and limited to second order terms:

$$
\begin{aligned}
\log[f_{\boldsymbol{\beta}}(\boldsymbol{\beta}\,|\,\mathbf{Y},\mathbf{X})] \quad \propto \quad & \log[f_{\boldsymbol{\beta}}(\boldsymbol{\beta}\,|\,\mathbf{Y},\mathbf{X})]|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{\mathrm{MAP}}} \\
& +\tfrac{1}{2}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}_{\mathrm{MAP}})^{\top}\ \frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\top}}\log[f_{\boldsymbol{\beta}}(\boldsymbol{\beta}\,|\,\mathbf{Y},\mathbf{X})]\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{\mathrm{MAP}}}\ (\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}_{\mathrm{MAP}})^{\top},
\end{aligned}
$$

in which the first order term cancels as the derivative of $f_{\boldsymbol{\beta}}(\boldsymbol{\beta}\,|\,\mathbf{Y},\mathbf{X})$ with respect to $\boldsymbol{\beta}$ vanishes at the posterior mode – its maximum. Take the exponential of this approximation and match its arguments to that of a multivariate Gaussian $\exp[-\tfrac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu}_\beta)^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}_\beta)]$. The covariance of the sought Gaussian approximation is thus the inverse of the Hessian of the negative penalized log-likelihood. Put together the posterior is approximated by:

$$
\boldsymbol{\beta}\,|\,\mathbf{Y},\mathbf{X} \sim \mathcal{N}\Big(\hat{\boldsymbol{\beta}}_{\mathrm{MAP}}, \Big\{\boldsymbol{\Delta} + \sum_{i=1}^{n}\frac{\exp(\mathbf{X}_i\boldsymbol{\beta})}{[1+\exp(\mathbf{X}_i\boldsymbol{\beta})]^2}\mathbf{X}_i\mathbf{X}_i^{\top}\Big\}^{-1}\Big).
$$

The Gaussian approximation is convenient but need not be good. Fortunately, the Bernstein-Von Mises Theorem (Van der Vaart, 2000) tells it is very accurate when the model is regular, the prior smooth, and the sample size sufficiently large. The quality of the approximation for an artificial example data set is shown in Figure 3.4.

3.5 Penalty parameter selection

As before the penalty parameter may be chosen through K-fold cross-validation. For the $K = n$ case Meijer and Goeman (2013) describe a computationally efficient approximation of the leave-one-out cross-validated loglikelihood. It is based on the exact evaluation of the LOOCV loss, discussed in Section 1.9.2, that avoided resampling. The approach of Meijer and Goeman (2013) hinges upon the first-order Taylor

expansion of the left-out penalized loglikelihood of the left-out estimate $\hat{\boldsymbol{\beta}}_{-i}(\lambda)$ around $\hat{\boldsymbol{\beta}}(\lambda)$, which yields an approximation of the former:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}_{-i}(\lambda) &\approx \hat{\boldsymbol{\beta}}(\lambda) - \left(\left. \frac{\partial^2 \mathcal{L}_{-i}^{\mathrm{pen}}}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} \right|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda)} \right)^{-1} \left. \frac{\partial \mathcal{L}_{-i}^{\mathrm{pen}}}{\partial \boldsymbol{\beta}} \right|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda)} \\
&= \hat{\boldsymbol{\beta}}(\lambda) + (\mathbf{X}_{-i,*}^\top \mathbf{W}_{-i,-i} \mathbf{X}_{-i,*} + \lambda \mathbf{I}_{pp})^{-1} \{\mathbf{X}_{-i,*}^\top [\mathbf{Y}_{-i} - \vec{\mathbf{g}}^{-1}(\mathbf{X}_{-i,*}; \hat{\boldsymbol{\beta}}(\lambda))] - \lambda \hat{\boldsymbol{\beta}}(\lambda)\}.
\end{aligned}
$$

This approximation involves the inverse of a $p \times p$ dimensional matrix, which amounts to the evaluation of n such inverses for the LOOCV loss. As in Section 1.9.2 this may be avoided. Rewrite both the gradient and the Hessian of the left-out loglikelihood in the approximation of the preceding display:

$$
\begin{aligned}
\mathbf{X}_{-i,*}^\top \{\mathbf{Y}_{-i} - \vec{\mathbf{g}}^{-1}(\mathbf{X}_{-i,*}; \hat{\boldsymbol{\beta}}(\lambda))]\} - \lambda \hat{\boldsymbol{\beta}}(\lambda) &= \mathbf{X}^\top \{\mathbf{Y} - \vec{\mathbf{g}}^{-1}[\mathbf{X}; \hat{\boldsymbol{\beta}}(\lambda)]\} - \lambda \hat{\boldsymbol{\beta}}(\lambda) - \mathbf{X}_{i,*}^\top \{Y_i - g^{-1}[\mathbf{X}_{i,*}; \hat{\boldsymbol{\beta}}(\lambda)]\} \\
&= -\mathbf{X}_{i,*}^\top \{Y_i - g^{-1}[\mathbf{X}_{i,*}; \hat{\boldsymbol{\beta}}(\lambda)]\}
\end{aligned}
$$

and

$$
\begin{aligned}
(\mathbf{X}_{-i,*}^\top \mathbf{W}_{-i,-i} \mathbf{X}_{-i,*} + \lambda \mathbf{I}_{pp})^{-1} &= (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} + \mathbf{W}_{ii} (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}_{i,*}^\top \\
&\quad [1 - \mathbf{H}_{ii}(\lambda)]^{-1} \mathbf{X}_{i,*} (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1},
\end{aligned}
$$

where the Woodbury identity has been used and now $\mathbf{H}_{ii}(\lambda) = \mathbf{W}_{ii} \mathbf{X}_{i,*} (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}_{i,*}^\top$. Substitute both in the approximation of the left-out ridge logistic regression estimator and manipulate as in Section 1.9.2 to obtain:

$$
\hat{\boldsymbol{\beta}}_{-i}(\lambda) \approx \hat{\boldsymbol{\beta}}(\lambda) - (\mathbf{X}^\top \mathbf{W} \mathbf{X} + \lambda \mathbf{I}_{pp})^{-1} \mathbf{X}_{i,*}^\top [1 - \mathbf{H}_{ii}(\lambda)]^{-1} [Y_i - g^{-1}(\mathbf{X}_{i,*}; \hat{\boldsymbol{\beta}}(\lambda))].
$$

Hence, the leave-one-out cross-validated loglikelihood $\sum_{i=1}^n \mathcal{L}[Y_i \mid \mathbf{X}_{i,*}, \hat{\boldsymbol{\beta}}_{-i}(\lambda)]$ can now be evaluated by means of a single inverse of a $p \times p$ dimensional matrix and some matrix multiplications. For the performance of this approximation in terms of accuracy and speed confer Meijer and Goeman (2013).

3.6 Application

The ridge logistic regression is used here to explain the status (dead or alive) of ovarian cancer samples at the close of the study from gene expression data at baseline. Data stem from the TCGA study (Cancer Genome Atlas Network, 2011), which measured gene expression by means of sequencing technology. Available are 295 samples with both status and transcriptomic profiles. These profiles are composed of 19990 transcript reads. The sequencing data, being representative of the mRNA transcript count, is heavily skewed. Zwiener *et al.* (2014) show that a simple transformation of the data prior to model building generally yields a better model than tailor-made approaches. Motivated by this observation the data were – to accommodate the zero counts – asinh-transformed. The logistic regression model is then fitted in ridge penalized fashion, leaving the intercept unpenalized. The ridge penalty parameter is chosen through 10-fold cross-validation minimizing the cross-validated error. R-code, and that for the sequel of this example, is to be found below.

Listing 3.1 R code

```
# load libraries
library(glmnet)
library(TCGA2STAT)

# load data
OVdata <- getTCGA(disease="OV", data.type="RNASeq", type="RPKM", clinical=TRUE)
Y      <- as.numeric(OVdata[[3]][,2])
X      <- asinh(data.matrix(OVdata[[3]][,-c(1:3)]))

# start fit
# optimize penalty parameter
cv.fit   <- cv.glmnet(X, Y, alpha=0, family=c("binomial"),
                          nfolds=10, standardize=FALSE)
```

```r
optL2      <- cv.fit$lambda.min

# estimate model
glmFit     <- glmnet(X, Y, alpha=0, family=c("binomial"),
                           lambda=optL2, standardize=FALSE)

# construct linear predictor and predicted probabilities
linPred  <- as.numeric(glmFit$a0 + X %*% glmFit$beta)
predProb <- exp(linPred) / (1+exp(linPred))

# visualize fit
boxplot(linPred ~ Y, pch=20, border="lightblue", col="blue",
                     ylab="linear␣predictor", xlab="response",
                     main="fit")

# evaluate predictive performance
# generate k-folds balanced w.r.t. status
fold       <- 10
folds1     <- rep(1:fold, ceiling(sum(Y)/fold))[1:sum(Y)]
folds0     <- rep(1:fold, ceiling((length(Y)-length(folds1))
                                  /fold))[1:(length(Y)-length(folds1))]
shuffle1 <- sample(1:length(folds1), length(folds1))
shuffle0 <- sample(1:length(folds0), length(folds0))
folds1   <- split(shuffle1, as.factor(folds1))
folds0   <- split(shuffle0, as.factor(folds0))
folds    <- list()
for (f in 1:fold){
  folds[[f]] <- c(which(Y==1)[folds1[[f]]], which(Y==0)[folds0[[f]]])
}
for (f in 1:fold){
  print(sum(Y[folds[[f]]]))
}

# build model
pred2obsL2 <- matrix(nrow=0, ncol=4)
colnames(pred2obsL2) <- c("optLambda", "linPred", "predProb", "obs")
for (f in 1:length(folds)){
  print(f)
  cv.fit     <- cv.glmnet(X[-folds[[f]],], Y[-folds[[f]]], alpha=0,
                          family=c("binomial"), nfolds=10, standardize=FALSE)
  optL2      <- cv.fit$lambda.min
  glmFit     <- glmnet(X[-folds[[f]],], Y[-folds[[f]]], alpha=0,
                     family=c("binomial"), lambda=optL2, standardize=FALSE)
  linPred  <- glmFit$a0 + X[folds[[f]],,drop=FALSE] %*% glmFit$beta
  predProb <- exp(linPred) / (1+exp(linPred))
  pred2obsL2 <- rbind(pred2obsL2, cbind(optL2, linPred, predProb, Y[folds[[f]]]
      ))
}

# visualize fit
boxplot(pred2obsL2[,3] ~ pred2obsL2[,4], pch=20, border="lightblue", col="blue"
    ,
                                        ylab="linear␣predictor", xlab="
                                            response",
                                        main="prediction")
```

The fit of the resulting model is studied. Hereto the fitted linear predictor $\mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_{\mathrm{opt}})$ is plotted against the status (Figure 3.5, left panel). The plot shows some overlap between the boxes, but also a clear separation. The latter suggests gene expression at baseline thus enables us to distinguish surviving from the to-be-diseased ovarian cancer patients. Ideally, a decision rule based on the linear predictor can be formulated to predict an individual's outcome.
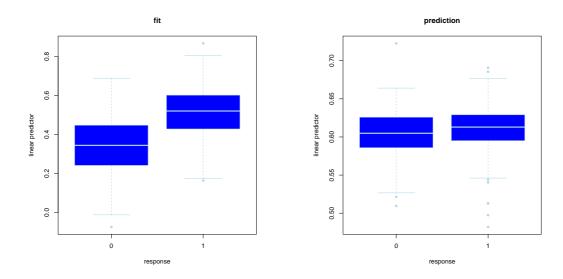
Figure 3.5: Left panel: Box plot of the status vs. the fitted linear predictor using the full data set. Right panel: Box plot of the status vs. the linear prediction in the left-out samples of the 10 folds.

The fit, however, is evaluated on the samples that have been used to build the model. This gives no insight on the model's predictive performance on novel samples. A replication of the study is generally costly and comparable data sets need not be at hand. A common workaround is to evaluate the predictive performance on the same data (Subramanian and Simon, 2010). This requires to put several samples aside for performance evaluation while the remainder is used for model building. The left-out sample may accidently be chosen to yield an exaggerated (either dramatically poor or overly optimistic) performance. This is avoided through the repetition of this exercise, leaving (groups of) samples out one at the time. The left-out performance evaluations are then averaged and believed to be representative of the predictive performance of the model on novel samples. Note that, effectively, as the model building involves cross-validation and so does the performance evaluation, a double cross-validation loop is applied. This procedure is applied with a ten-fold split in both loops. Denote the outer folds by $f = 1, \ldots, 10$. Then, $\mathbf{X}_f$ and $\mathbf{X}_{-f}$ represent the design matrix of the samples comprising fold f and that of the remaining samples, respectively. Define $\mathbf{Y}_f$ and $\mathbf{Y}_{-f}$ similarly. The linear prediction for the left-out fold f is then $\mathbf{X}_f \hat{\boldsymbol{\beta}}_{-f}(\lambda_{\mathrm{opt, -f}})$. For reference to the fit, this is compared to $\mathbf{Y}_f$ visually by means of a boxplot as used above (see Figure 3.5, right panel). The boxes overlap almost perfectly. Hence, little to nothing remains of the predictive power suggested by the boxplot of the fit. The fit may thus give a reasonable description of the data at hand, but it extrapolates poorly to new samples.

3.7 Conclusion

To deal with response variables other than continuous ones, ridge logistic regression was discussed. High-dimensionally, the empirical identifiability problem then persists. Again, penalization came to the rescue: the ridge penalty may be combined with other link functions than the identity. Properties of ridge regression were shown to carry over to its logistic equivalent.

3.8 Exercises

Question 3.1
Consider an experiment involving n cancer samples. For each sample i the transcriptome of its tumor has been profiled and is denoted $\mathbf{X}_i = (X_{i1}, \ldots, X_{ip})^\top$ where X_{ij} represents the gene $j = 1, \ldots, p$ in sample i. Additionally, the overall survival data, (Y_i, c_i) for $i = 1, \ldots, n$ of these samples is available. In this Y_i denotes the survival time of sample i and c_i the event indicator with $c_i = 0$ and $c_i = 1$ representing non- and censoredness, respectively. You may ignore the possibility of ties in the remainder.

a) Write down the Cox proportional regression model that links overall survival times (as the response variable) to the expression levels.
b) Specify its loss function for penalized maximum partial (!) likelihood estimation of the parameters. Penalization is via the ridge penalty.
c) From this loss function, derive the estimation equation for the Cox regression coefficients.
d) Describe (in words) how you would find the 'ridge ML estimate'.

Question 3.2
Download the `multtest` package from BioConductor:
```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("multtest")
```
Activate the library and load leukemia data from the package:
```
> library(multtest)
> data(golub)
```
The objects `golub` and `golub.cl` are now available. The matrix-object `golub` contains the expression profiles of 38 leukemia patients. Each profile comprises expression levels of 3051 genes. The numeric-object `golub.cl` is an indicator variable for the leukemia type (AML or ALL) of the patient.

a) Relate the leukemia subtype and the gene expression levels by a logistic regression model. Fit this model by means of penalized maximum likelihood, employing the ridge penalty with penalty parameter $\lambda = 1$. This is implemented in the `penalized`-packages available from `CRAN`. *Note:* center (gene-wise) the expression levels around zero.
b) Obtain the fits from the regression model. The fit is almost perfect. Could this be due to overfitting the data? Alternatively, could it be that the biological information in the gene expression levels indeed determines the leukemia subtype almost perfectly?
c) To discern between the two explanations for the almost perfect fit, randomly shuffle the subtypes. Refit the logistic regression model and obtain the fits. On the basis of this and the previous fit, which explanation is more plausible?
d) Compare the fit of the logistic model with different penalty parameters, say $\lambda = 1$ and $\lambda = 1000$. How does λ influence the possibility of overfitting the data?
e) Describe what you would do to prevent overfitting.

Question 3.3
Download the `breastCancerNKI` package from BioConductor:
```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("breastCancerNKI")
```
Activate the library and load leukemia data from the package:
```
> library(breastCancerNKI)
> data(nki)
```
The eset-object `nki` is now available. It contains the expression profiles of 337 breast cancer patients. Each profile comprises expression levels of 24481 genes. Extract the expression data from the object, remove all genes with missing values, center the gene expression gene-wise around zero, and limit the data set to the first thousand genes. The reduction of the gene dimensionality is only for computational speed.
```
X <- exprs(nki)
X <- X[-which(rowSums(is.na(X)) > 0),]
X <- apply(X[1:1000,], 1, function(X) X - mean(X) ) .
```
Furthermore, extract the estrogen receptor status (short: ER status), an important prognostic indicator for breast cancer.
```
Y <- pData(nki)[,8]
```
a) Relate the ER status and the gene expression levels by a logistic regression model, which is fitted by means of ridge penalized maximum likelihood. First, find the optimal value of the penalty parameter of λ by means of cross-validation. This is implemented in `optL2`-function of the `penalized`-package available from `CRAN`.
b) Evaluate whether the cross-validated likelihood indeed attains a maximum at the optimal value of λ. This can be done with the `profL2`-function of the `penalized`-package available from `CRAN`.
c) Investigate the sensitivity of the penalty parameter selection with respect to the choice of the cross-validation fold.

d) Does the optimal lambda produce a reasonable fit?

4 Lasso regression

In this chapter we return to the linear regression model, which is still fitted in penalized fashion but this time with a so-called lasso penalty. Yet another penalty? Yes, but it will turn out to have interesting consequences. The outline of this chapter loosely follows that of its counterpart on ridge regression (Chapter 1). The chapter can – at least partially – be seen as an elaborated version of the original work on lasso regression, i.e. Tibshirani (1996), with most topics covered and visualized more extensively and incorporating results and examples published since.

Recall that ridge regression finds an estimator of the parameter the linear regression model through the minimization of:

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + f_{\text{pen}}(\boldsymbol{\beta}, \lambda), \tag{4.1}$$

with $f_{\text{pen}}(\boldsymbol{\beta}, \lambda) = \lambda\|\boldsymbol{\beta}\|_2^2$. The particular choice of the penalty function originated in a post-hoc motivation of the ad-hoc fix to the singularity of the matrix $\mathbf{X}^\top\mathbf{X}$, stemming from the design matrix $\mathbf{X}$ not being of full rank (i.e. $\text{rank}(\mathbf{X}) < p$). The ad-hoc nature of the fix suggests that the choice for the squared Euclidean norm of $\boldsymbol{\beta}$ as a penalty is somewhat arbitrary and other choices may be considered, some of which were already encountered in Chapter 2.

One such choice is the so-called lasso penalty giving rise to lasso regression, as introduced by Tibshirani (1996). Like ridge regression, lasso regression fits the linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with the standard assumption on the error $\boldsymbol{\varepsilon}$. Like ridge regression, it does so by minimizing the sum of squares augmented with a penalty. Hence, lasso regression too minimizes loss function (4.1). The difference with ridge regression is in the penalty function. Instead of the squared Euclidean norm, lasso regression uses the ℓ_1-norm: $f_{\text{pen}}(\boldsymbol{\beta}, \lambda_1) = \lambda_1\|\boldsymbol{\beta}\|_1$, the sum of the absolute values of the regression parameters multiplied by the lasso penalty parameter λ_1. To distinguish the ridge and lasso penalty parameters they are henceforth denoted λ_2 and λ_1, respectively, with the subscript referring to the norm used in the penalty. The lasso regression loss function is thus:

$$\mathcal{L}_{\text{lasso}}(\boldsymbol{\beta}; \lambda) \;=\; \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \;=\; \sum_{i=1}^{n}(Y_i - \mathbf{X}_{i*}\boldsymbol{\beta})^2 + \lambda_1\sum_{j=1}^{p}|\beta_j|. \tag{4.2}$$

The lasso regression estimator is then defined as the minimizer of this loss function. As with the ridge regression loss function, the maximum likelihood estimate of $\boldsymbol{\beta}$ minimizes the first part, and second part is minimized by setting $\boldsymbol{\beta}$ equal to the p dimensional zero vector. For λ_1 close to zero, the lasso estimate is close to the maximum likelihood estimate. Whereas for large λ_1, the penalty term overshadows the sum-of-squares, and the lasso estimate is small (in some sense). Intermediate choices of λ_1 mold a compromise between those two extremes, with the penalty parameter determining the contribution of each part to this compromise. The lasso regression estimator thus is not one but a whole sequence of estimators of $\boldsymbol{\beta}$, one for every $\lambda_1 \in \mathbb{R}_{>0}$. This sequence is the lasso regularization path, defined as $\{\hat{\boldsymbol{\beta}}(\lambda_1) : \lambda_1 \in \mathbb{R}_{>0}\}$. To arrive at a final lasso estimator of $\boldsymbol{\beta}$, like its ridge counterpart, the lasso penalty parameter λ_1 needs to be chosen.

The ℓ_1 penalty of lasso regression is equally arbitrary as the ℓ_2-penalty of ridge regression. The latter ensured the existence of a well-defined estimator of the regression parameter $\boldsymbol{\beta}$ in the presence of super-collinearity in the design matrix $\mathbf{X}$, in particular when the dimension p exceeds the sample size n. The augmentation of the sum-of-squares with the lasso penalty achieves the same. This is illustrated in Figure 4.1. For the high-dimensional setting with $p = 2$ and $n = 1$ and arbitrary data the level sets of the sum-of-squares $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ and the lasso regression loss $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1$ are plotted (left and right panel, respectively). In both panels the minimum is indicated in red. For the sum-of-squares the
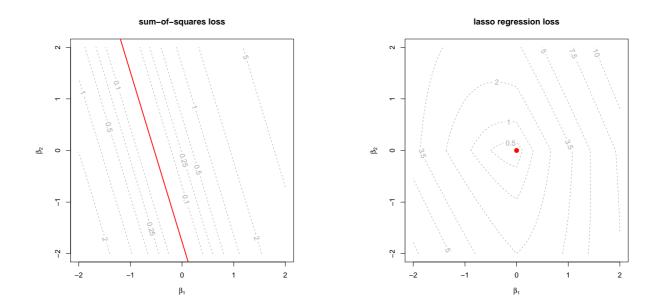
Figure 4.1: Contour plots of the sum-of-squares and the lasso regression loss (left and right panel, respectively). The dotted grey line represent level sets. The red line and dot represent the the location of minimum in both panels.

minimum is a line. As pointed out before in Section 1.2 of Chapter 1 on ridge regression, this minimum is determined up to an element of the null set of the design matrix $\mathbf{X}$, which in this case is non-trivial. In contrast, the lasso regression loss exhibits a unique well-defined minimum. Hence, the augmentation of the sum-of-squares with the lasso penalty yields a well-defined estimator of the regression parameter. (This needs some attenuation: in general the minimum of the lasso regression loss need not be unique, confer Section 4.1).

The mathematics involved in the derivation in this chapter tends to be more intricate than for ridge regression. This is due to the non-differentiability of the lasso penalty at zero. This has consequences on all aspects of the lasso regression estimator as is already obvious in the left-hand panel of Figure 4.1: confer the cusps in the lasso regression loss level sets.

4.1   Uniqueness

The lasso regression loss function is the sum of the sum-of-squares criterion and a sum of absolute value functions. Both are convex in $\boldsymbol{\beta}$: the former is not strict convex due to the high-dimensionality and the absolute value function is convex due to its piece-wise linearity. Thereby the lasso loss function too is convex but not strict. Consequently, its minimum need not be uniquely defined. But, the set of solutions of a convex minimization problem is convex (Theorem 9.4.1, Fletcher, 1987). Hence, would there exist multiple minimizers of the lasso loss function, they form a convex set. Thus, if $\hat{\boldsymbol{\beta}}_a(\lambda_1)$ and $\hat{\boldsymbol{\beta}}_b(\lambda_1)$ are lasso estimators, then so are $(1 - \theta)\hat{\boldsymbol{\beta}}_a(\lambda_1) + \theta\hat{\boldsymbol{\beta}}_b(\lambda_1)$ for $\theta \in (0, 1)$. This is illustrated in Example 4.1.

Example 4.1 *(Perfectly super-collinear covariates)*
Consider the standard linear regression model $Y_i = \mathbf{X}_{i,*}\boldsymbol{\beta} + \varepsilon_i$ for $i = 1, \ldots, n$ and with the ε_i i.i.d. normally distributed with zero mean and a common variance. The rows of the design matrix $\mathbf{X}$ are of length two, neither column represents the intercept, but $\mathbf{X}_{*,1} = \mathbf{X}_{*,2}$. Suppose an estimate of the regression parameter $\boldsymbol{\beta}$ of this model is obtained through the minimization of the sum-of-squares augmented with a lasso penalty, $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1$ with penalty parameter $\lambda_1 > 0$. To find the minimizer define $u = \beta_1 + \beta_2$ and $v = \beta_1 - \beta_2$ and rewrite the lasso loss criterion to:

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \;\; = \;\; \|\mathbf{Y} - \mathbf{X}_{*,1}u\|_2^2 + \tfrac{1}{2}\lambda_1(|u + v| + |u - v|).$$

The function $|u + v| + |u - v|$ is minimized with respect to v for any v such that $|v| < |u|$ and the

corresponding minimum equals $2|u|$. The estimator of u thus minimizes:

$$\|\mathbf{Y} - \mathbf{X}_{*,1}u\|_2^2 + \lambda_1|u|.$$

For sufficiently small values of λ_1 the estimate of u will be unequal to zero. Then, any v such that $|v| < |u|$ will yield the same minimum of the lasso loss function. Consequently, $\hat{\boldsymbol{\beta}}(\lambda_1)$ is not uniquely defined as $\hat{\beta}_1(\lambda_1) = \frac{1}{2}[\hat{u}(\lambda_1) + \hat{v}(\lambda_1)]$ need not equal $\hat{\beta}_2(\lambda_1) = \frac{1}{2}[\hat{u}(\lambda_1) - \hat{v}(\lambda_1)]$ for any $\hat{v}(\lambda_1)$ such that $0 < |\hat{v}(\lambda_1)| < |\hat{u}(\lambda_1)|$. $\qquad\square$

The lasso estimator $\hat{\boldsymbol{\beta}}(\lambda_1)$ may not be unique, but its linear predictor $\mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1)$ is. This can be proven by contradiction (Tibshirani, 2013). Suppose there exists two lasso estimators of $\boldsymbol{\beta}$, denoted $\hat{\boldsymbol{\beta}}_a(\lambda_1)$ and $\hat{\boldsymbol{\beta}}_b(\lambda_1)$, such that $\mathbf{X}\hat{\boldsymbol{\beta}}_a(\lambda_1) \neq \mathbf{X}\hat{\boldsymbol{\beta}}_b(\lambda_1)$. Define c to be minimum of the lasso loss function. Then, by definition of the lasso estimators $\hat{\boldsymbol{\beta}}_a(\lambda_1)$ and $\hat{\boldsymbol{\beta}}_b(\lambda_1)$ satisfy:

$$\|\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_a(\lambda_1)\|_2^2 + \lambda_1\|\hat{\boldsymbol{\beta}}_a(\lambda_1)\|_1 \;=\; c \;=\; \|\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_b(\lambda_1)\|_2^2 + \lambda_1\|\hat{\boldsymbol{\beta}}_b(\lambda_1)\|_1.$$

For $\theta \in (0,1)$ we then have:

$$\begin{aligned}
&\|\mathbf{Y} - \mathbf{X}[(1-\theta)\hat{\boldsymbol{\beta}}_a(\lambda_1) + \theta\hat{\boldsymbol{\beta}}_b(\lambda_1)]\|_2^2 + \lambda_1\|(1-\theta)\hat{\boldsymbol{\beta}}_a(\lambda_1) + \theta\hat{\boldsymbol{\beta}}_b(\lambda_1)\|_1 \\
&= \; \|(1-\theta)[\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_a(\lambda_1)] + \theta[\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_b(\lambda_1)]\|_2^2 + \lambda_1\|(1-\theta)\hat{\boldsymbol{\beta}}_a(\lambda_1) + \theta\hat{\boldsymbol{\beta}}_b(\lambda_1)\|_1 \\
&< \; (1-\theta)\|\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_a(\lambda_1)\|_2^2 + \theta\|\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}}_b(\lambda_1)\|_2^2 + (1-\theta)\lambda_1\|\hat{\boldsymbol{\beta}}_a(\lambda_1)\|_1 + \theta\lambda_1\|\hat{\boldsymbol{\beta}}_b(\lambda_1)\|_1 \\
&= \; (1-\theta)c + \theta c \;=\; c,
\end{aligned}$$

by the strict convexity of $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ in $\mathbf{X}\boldsymbol{\beta}$ and the convexity of $\|\boldsymbol{\beta}\|_1$ on $\theta \in (0,1)$. This implies that $(1-\theta)\hat{\boldsymbol{\beta}}_a(\lambda_1) + \theta\hat{\boldsymbol{\beta}}_b(\lambda_1)$ yields a lower minimum of the lasso loss function and contradicts our assumption that $\hat{\boldsymbol{\beta}}_a(\lambda_1)$ and $\hat{\boldsymbol{\beta}}_b(\lambda_1)$ are lasso regression estimators.

Example 4.2 *(Perfectly super-collinear covariates, revisited)*
Revisit the setting of Example 4.1, where a linear regression model without intercept and only two but perfectly correlated covariates is fitted to data. The example revealed that the lasso estimator need not be unique. The lasso predictor, however, is

$$\widehat{\mathbf{Y}}(\lambda_1) \;=\; \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1) \;=\; \mathbf{X}_{*,1}\hat{\beta}_1(\lambda_1) + \mathbf{X}_{*,2}\hat{\beta}_2(\lambda_1) \;=\; \mathbf{X}_{*,1}[\hat{\beta}_1(\lambda_1) + \hat{\beta}_2(\lambda_1)] \;=\; \mathbf{X}_{*,1}\hat{u}(\lambda_1),$$

with u defined and (uniquely) estimated as in Example 4.1. $\qquad\square$

Example 4.3
The issues, non- and uniqueness of the lasso-estimator and predictor, respectively, raised above are illustrated in a numerical setting. Hereto data are generated in accordance with the linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where the $n = 5$ rows of $\mathbf{X}$ are sampled from $\mathcal{N}[\mathbf{0}_p, (1-\rho)\mathbf{I}_{pp} + \rho\mathbf{1}_{pp}]$ with $p = 10$, $\rho = 0.99$, $\boldsymbol{\beta} = (\mathbf{1}_3^\top, \mathbf{0}_{p-3}^\top)^\top$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_p, \frac{1}{10}\mathbf{I}_{nn})$. With these data the lasso estimator of the regression parameter $\boldsymbol{\beta}$ for $\lambda_1 = 1$ is evaluated using two different algorithms (see Section 4.4). Employed implementations of the algorithms are those available through the R-packages `penalized`- and `glmnet`. Both estimates, denoted $\hat{\boldsymbol{\beta}}_{\mathtt{p}}(\lambda_1)$ and $\hat{\boldsymbol{\beta}}_{\mathtt{g}}(\lambda_1)$ (the subscript refers to the first letter of the package), are given in Table 4.3. The table reveals that the estimates differ, in particular in their support (i.e. the set of nonzero values of

		β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
`penalized`	$\hat{\boldsymbol{\beta}}_{\mathtt{p}}(\lambda_1)$	0.267	0.000	1.649	0.093	0.000	0.000	0.000	0.571	0.000	0.269
`glmnet`	$\hat{\boldsymbol{\beta}}_{\mathtt{g}}(\lambda_1)$	0.269	0.000	1.776	0.282	0.195	0.000	0.000	0.325	0.000	0.000

Table 4.1: Lasso estimates of the linear regression $\boldsymbol{\beta}$ for both algorithms.

the estimate of $\boldsymbol{\beta}$). This is troublesome when it comes to communication of the optimal model. From a different perspective the realized loss $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1$ for each estimate is approximately equal 2.99, with the difference possibly due to convergence criteria of the algorithms. On another note, their corresponding predictors, $\mathbf{X}\hat{\boldsymbol{\beta}}_{\mathtt{p}}(\lambda_1)$ and $\mathbf{X}\hat{\boldsymbol{\beta}}_{\mathtt{g}}(\lambda_1)$, correlate almost perfectly: $\mathrm{cor}[\mathbf{X}\hat{\boldsymbol{\beta}}_{\mathtt{p}}(\lambda_1), \mathbf{X}\hat{\boldsymbol{\beta}}_{\mathtt{g}}(\lambda_1)] = 0.999$. These results thus corroborate the non-uniqueness of the estimator and the uniqueness of the predictor.

The R-script provides the code to reproduce the analysis.

Listing 4.1 R code

```r
# set the random seed
set.seed(4)

# load libraries
library(penalized)
library(glmnet)
library(mvtnorm)

# set sample size
p <- 10

# create covariance matrix
Sigma <- matrix(0.99, p, p)
diag(Sigma) <- 1

# sample the design matrix
n <- 5
X <- rmvnorm(10, sigma=Sigma)

# create a sparse beta vector
betas <- c(rep(1, 3), rep(0, p-3))

# sample response
Y <- X %*% betas + rnorm(n, sd=0.1)

# evaluate lasso estimator with two methods
Bhat1 <- matrix(as.numeric(coef(penalized(Y, X, lambda1=1, unpenalized=~0),
                                "all")), ncol=1)
Bhat2 <- matrix(as.numeric(coef(glmnet(X, Y, lambda=1/(2*n), standardize=FALSE,
                                intercept=FALSE)))[-1], ncol=1)

# compare estimates
cbind(Bhat1, Bhat2)

# compare the loss
sum((Y - X %*% Bhat1)^2) + sum(abs(Bhat1))
sum((Y - X %*% Bhat2)^2) + sum(abs(Bhat2))

# compare predictor
cor(X %*% Bhat1, X %*% Bhat2)
```

Note that in the code above the evaluation of the lasso estimator appears to employ a different lasso penalty parameter λ_1. This is due to the fact that internally (after removal of standardization of $\mathbf{X}$ and $\mathbf{Y}$) the loss functions optimized are $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1$ vs. $\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1$. Rescaling of λ_1 resolves this issue. $\square$

4.2 Analytic solutions

In general, no explicit expression for the lasso regression estimator exists. There are exceptions, as illustrated in Examples 4.4 and 4.6. Nonetheless, it is possible to show properties of the lasso estimator, amongst others of the smoothness of its regularization path (Theorem 4.1) and the limiting behaviour as $\lambda_1 \to \infty$ (see the end of this section).
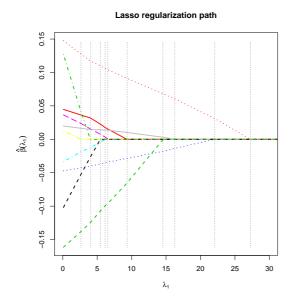
Theorem 4.1 (Theorem 2, Rosset and Zhu, 2007)
The lasso regression loss function (4.2) yields a piecewise linear (in λ_1) regularization path $\{\hat{\boldsymbol{\beta}}(\lambda_1) : \mathbb{R}_{>0}\}$.

Proof Confer Rosset and Zhu (2007). ∎

This piecewise linear nature of the lasso solution path is illustrated in the left-hand panel of Figure 4.2 of an arbitrary data set. At each vertical dotted line a discontinuity in the derivative with respect to λ_1

of the regularization path of an lasso estimate of an element of $\boldsymbol{\beta}$ may occur. The plot also foreshadows the $\lambda_1 \to \infty$ limiting behaviour of the lasso regression estimator: the estimator tend to zero. This is no surprise knowing that the ridge regression estimator exhibits the same behaviour and the lasso regression loss function is of similar form as that of ridge regression: a sum-of-squares plus a penalty term (which is linear in the penalty parameter).
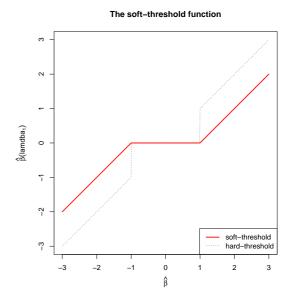
Lasso regularization path   **The soft–threshold function**

Figure 4.2: The left panel shows the regularization path of the lasso regression estimator for simulated data. The vertical grey dotted lines indicate the values of λ_1 at which there is a discontinuity in the derivative (with respect to λ_1) of the lasso regularization path of one the regression estimates. The right panel displays the soft (solid, red) and hard (grey, dotted) threshold functions.

For particular cases, an orthormal design (Example 4.4) and $p = 2$ (Example 4.6), an analytic expression for the lasso regression estimator exists. While the latter is of limited use, the former is exemplary and will come of use later in the numerical evaluation of the lasso regression estimator in the general case (see Section 4.4).

Example 4.4 *Orthonormal design matrix*
Consider an orthonormal design matrix $\mathbf{X}$, i.e. $\mathbf{X}^\top \mathbf{X} = \mathbf{I}_{pp} = (\mathbf{X}^\top \mathbf{X})^{-1}$. The lasso estimator then is:

$$\hat{\beta}_j(\lambda_1) \;\; = \;\; \operatorname{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \tfrac{1}{2}\lambda_1)_+,$$

where $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y} = \mathbf{X}^\top \mathbf{Y}$ is the maximum likelihood estimator of $\boldsymbol{\beta}$ and $\hat{\beta}_j$ its j-th element and $f(x) = (x)_+ = max\{x, 0\}$. This expression for the lasso regression estimator can be obtained as follows. Rewrite the lasso regression loss criterion:

$$\begin{aligned}
\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 \;\; &= \;\; \min_{\boldsymbol{\beta}} \mathbf{Y}^\top \mathbf{Y} - \mathbf{Y}^\top \mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{Y} + \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} + \lambda_1 \sum_{j=1}^{p} |\beta_j| \\[2mm]
&\propto \;\; \min_{\boldsymbol{\beta}} -\hat{\boldsymbol{\beta}}^\top \boldsymbol{\beta} - \boldsymbol{\beta}^\top \hat{\boldsymbol{\beta}} + \boldsymbol{\beta}^\top \boldsymbol{\beta} + \lambda_1 \sum_{j=1}^{p} |\beta_j| \\[2mm]
&= \;\; \min_{\beta_1,\dots,\beta_p} \sum_{j=1}^{p} \big(-2\hat{\beta}_j^{\text{OLS}} \beta_j + \beta_j^2 + \lambda_1 |\beta_j| \big) \\[2mm]
&= \;\; \sum_{j=1}^{p} \big(\min_{\beta_j} -2\hat{\beta}_j \beta_j + \beta_j^2 + \lambda_1 |\beta_j| \big).
\end{aligned}$$

The minimization problem can thus be solved per regression coefficient. This gives:

$$\min_{\beta_j} -2\hat{\beta}_j\,\beta_j + \beta_j^2 + \lambda_1|\beta_j| \;\; = \;\; \begin{cases} \min_{\beta_j} -2\hat{\beta}_j\,\beta_j + \beta_j^2 + \lambda_1\beta_j & \text{if} \quad \beta_j > 0, \\ \min_{\beta_j} -2\hat{\beta}_j\,\beta_j + \beta_j^2 - \lambda_1\beta_j & \text{if} \quad \beta_j < 0. \end{cases}$$

The minimization within the sum over the covariates is with respect to each element of the regression parameter separately. Optimization with respect to the j-th one gives:

$$\hat{\beta}_j(\lambda_1) \;\; = \;\; \begin{cases} \hat{\beta}_j - \frac{1}{2}\lambda_1 & \text{if} \quad \beta_j > 0 \\ \hat{\beta}_j + \frac{1}{2}\lambda_1 & \text{if} \quad \beta_j < 0 \end{cases}$$

Put these two equations together to arrive at the form of the lasso regression estimator above.

The analytic expression for the lasso regression estimator above provides insight in how it relates to the maximum likelihood estimator of $\boldsymbol{\beta}$. The right-hand side panel of Figure 4.2 depicts this relationship. Effectively, the lasso regression estimator thresholds (after a translation) its maximum likelihood counterpart. The function is also referred to as the *soft-threshold function* (for contrast the hard-threshold function is also plotted – dotted line – in Figure 4.2). □

Example 4.5 *(Orthogonal design matrix)*
The analytic solution of the lasso regression estimator for experiments with an orthonormal design matrix applies to those with an orthogonal design matrix. This is illustrated by a numerical example. Use the lasso estimator with $\lambda_1 = 10$ to fit the linear regression model to the response data and the design matrix:

$$\mathbf{Y}^\top \;\; = \;\; \begin{pmatrix} -4.9 & -0.8 & -8.9 & 4.9 & 1.1 & -2.0 \end{pmatrix},$$
$$\mathbf{X}^\top \;\; = \;\; \begin{pmatrix} 1 & -1 & 3 & -3 & 1 & 1 \\ -3 & -3 & -1 & 0 & 3 & 0 \end{pmatrix}.$$

Note that the design matrix is orthogonal, i.e. its columns are orthogonal (but not normalized to one). The orthogonality of $\mathbf{X}$ yields a diagonal $\mathbf{X}^\top\mathbf{X}$, and so it its inverse $(\mathbf{X}^\top\mathbf{X})^{-1}$. Here $\text{diag}(\mathbf{X}^\top\mathbf{X}) = (22, 28)$. Rescale $\mathbf{X}$ to an orthonormal design matrix, denoted $\tilde{\mathbf{X}}$, and rewrite the lasso regression loss function to:

$$\begin{aligned} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \;\; &= \;\; \left\| \mathbf{Y} - \mathbf{X} \begin{pmatrix} \sqrt{22} & 0 \\ 0 & \sqrt{28} \end{pmatrix}^{-1} \begin{pmatrix} \sqrt{22} & 0 \\ 0 & \sqrt{28} \end{pmatrix} \boldsymbol{\beta} \right\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \\ &= \;\; \|\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\gamma}\|_2^2 + (\lambda_1/\sqrt{22})|\gamma_1| + (\lambda_1/\sqrt{28})|\gamma_2|, \end{aligned}$$

where $\boldsymbol{\gamma} = (\sqrt{22}\beta_1, \sqrt{28}\beta_2)^\top$. By the same argument this loss can be minimized with respect to each element of $\boldsymbol{\gamma}$ separately. In particular, the soft-threshold function provides an analytic expression for the estimates of $\boldsymbol{\gamma}$:

$$\begin{aligned} \hat{\gamma}_1(\lambda_1/\sqrt{22}) \;\; &= \;\; \text{sign}(\hat{\gamma}_1)[|\hat{\gamma}_j| - \tfrac{1}{2}(\lambda_1/\sqrt{22})]_+ \;\; = \;\; -[9.892513 - \tfrac{1}{2}(10/\sqrt{22})]_+ \;\; = \;\; 8.826509, \\ \hat{\gamma}_1(\lambda_1/\sqrt{28}) \;\; &= \;\; \text{sign}(\hat{\gamma}_2)[|\hat{\gamma}_2| - \tfrac{1}{2}(\lambda_1/\sqrt{28})]_+ \;\; = \;\; [5.537180 - \tfrac{1}{2}(10/\sqrt{28})]_+ \;\; = \;\; 4.592269. \end{aligned}$$

Rescale back and obtain the lasso regression estimate: $\hat{\boldsymbol{\beta}}(10) = (-1.881818, 0.8678572)^\top$. □

Example 4.6 *(p = 2 with equivariant covariates, Leng* et al.*, 2006)*
Let $p = 2$ and suppose the design matrix $\mathbf{X}$ has equivariant covariates. Without of loss of generality they are assumed to have unit variance. We may thus write

$$\mathbf{X}^\top\mathbf{X} \;\; = \;\; \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},$$

for some $\rho \in (-1, 1)$. The lasso regression estimator is then of similar form as in the orthonormal case: $\hat{\beta}_j(\lambda_1) = \text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \gamma)_+$, with soft-threshold parameter γ that now depends on λ_1, ρ and the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ (see Exercise 4.3). □

Apart from the specific cases outlined in the two examples above no other explicit solutions for the minimizer of the lasso regression loss function appears to be known. Locally though, for large enough values of λ_1, an analytic expression for solution can also be derived. Hereto we point out that (details omitted) the lasso estimator satisfies the following estimating equation:

$$\mathbf{X}^\top \mathbf{X} \hat{\boldsymbol{\beta}}(\lambda_1) \;\; = \;\; \mathbf{X}^\top \mathbf{Y} - \tfrac{1}{2}\lambda_1 \hat{\mathbf{z}}$$

for some $\hat{\mathbf{z}} \in \mathbb{R}^p$ with $(\hat{\mathbf{z}})_j = \text{sign}\{[\hat{\boldsymbol{\beta}}(\lambda_1)]_j\}$ whenever $[\hat{\boldsymbol{\beta}}_j(\lambda_1)]_j \neq 0$ and $(\hat{\mathbf{z}})_j \in [-1, 1]$ if $[\hat{\boldsymbol{\beta}}_j(\lambda_1)]_j = 0$. Then:

$$0 \;\; \leq \;\; [\hat{\boldsymbol{\beta}}(\lambda_1)]^\top \mathbf{X}^\top \mathbf{X} \hat{\boldsymbol{\beta}}(\lambda_1) \;\; = \;\; [\hat{\boldsymbol{\beta}}(\lambda_1)]^\top (\mathbf{X}^\top \mathbf{Y} - \tfrac{1}{2}\lambda_1 \hat{\mathbf{z}}) \;\; = \;\; \sum_{j=1}^{p} [\hat{\boldsymbol{\beta}}(\lambda_1)]_j (\mathbf{X}^\top \mathbf{Y} - \tfrac{1}{2}\lambda_1 \hat{\mathbf{z}})_j.$$

For $\lambda_1 > 2\|\mathbf{X}^\top \mathbf{Y}\|_\infty$ the summands on the right-hand side satisfy:

$$\begin{array}{rcll}
[\hat{\boldsymbol{\beta}}(\lambda_1)]_j (\mathbf{X}^\top \mathbf{Y} - \tfrac{1}{2}\lambda_1 \hat{\mathbf{z}})_j & < & 0 & \text{if} \quad [\hat{\boldsymbol{\beta}}(\lambda_1)]_j > 0, \\
[\hat{\boldsymbol{\beta}}(\lambda_1)]_j (\mathbf{X}^\top \mathbf{Y} - \tfrac{1}{2}\lambda_1 \hat{\mathbf{z}})_j & = & 0 & \text{if} \quad [\hat{\boldsymbol{\beta}}(\lambda_1)]_j = 0, \\
[\hat{\boldsymbol{\beta}}(\lambda_1)]_j (\mathbf{X}^\top \mathbf{Y} - \tfrac{1}{2}\lambda_1 \hat{\mathbf{z}})_j & < & 0 & \text{if} \quad [\hat{\boldsymbol{\beta}}(\lambda_1)]_j < 0.
\end{array}$$

This implies that $\hat{\boldsymbol{\beta}}(\lambda_1) = \mathbf{0}_p$ if $\lambda_1 > 2\|\mathbf{X}^\top \mathbf{Y}\|_\infty$, where $\|\mathbf{a}\|_\infty$ is the supremum norm of vector $\mathbf{a}$ defined as $\|\mathbf{a}\|_\infty = \max\{|a_1|, |a_2|, \ldots, |a_p|\}$.

4.3 Sparsity

The change from the ℓ_2-norm to the ℓ_1-norm in the penalty may seem only a detail. Indeed, both ridge and lasso regression fit the same linear regression model. But the attractiveness of the lasso lies not in *what* it fits, but in a *consequence of how* it fits the linear regression model. The lasso estimator of the vector of regression parameters may contain some or many zero's. In contrast, ridge regression yields an estimator of $\boldsymbol{\beta}$ with elements (possibly) close to zero, but unlikely equal to zero. Hence, lasso penalization results in $\hat{\beta}_j(\lambda_1) = 0$ for some j (in particular for large values of λ_1, see Section 4.1), while ridge penalization yields an estimate of the j-th element of the regression parameter $\hat{\beta}_j(\lambda_2) \neq 0$. A zero estimate of a regression coefficient means that the corresponding covariate has no effect on the response and can be excluded from the model. Effectively, this amounts to variable selection. Where traditionally the linear regression model is fitted by means of maximum likelihood followed by testing step to weed out these covariates with effects indistinguishable from zero, lasso regression is a one-step-go procedure that simultaneously estimates and selects.

The in-built variable selection of the lasso regression estimator is a geometric accident. To understand how it comes about the lasso regression loss optimization problem (4.2) is reformulated as a constrained estimation problem (using the same argumentation as previously employed for ridge regression, see Section 1.5):

$$\min_{\|\boldsymbol{\beta}\|_1 \leq c(\lambda_1)} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2.$$

where $c(\lambda_1) = \|\hat{\boldsymbol{\beta}}(\lambda_1)\|_1$. Again, this is the standard least squares problem, with the only difference that the sum of the (absolute) regression parameters $\beta_1, \beta_2, \ldots, \beta_p$ is required to be smaller than $c(\lambda_1)$. The effect of this requirement is that the lasso estimates of the regression parameters can no longer assume any value (from $-\infty$ to ∞, as is the case in standard linear regression), but are limited to a certain range of values. With the lasso and ridge regression estimators minimizing the same sum-of-squares, the key difference with the constrained estimation formulation of ridge regression is not in the explicit form of $c(\lambda_1)$ (and is set to some arbitrary convenient value in the remainder of this section) but in what is bounded by $c(\lambda_1)$ and the domain of acceptable values for $\boldsymbol{\beta}$ that it implies. For the lasso regression estimator the domain is specified by a bound on the ℓ_1-norm of the regression parameter while for its ridge counterpart the bound is applied to the squared ℓ_2-norm of $\boldsymbol{\beta}$. The parameter constraints implied by the lasso and ridge norms result in balls in different norms:

$$\{\boldsymbol{\beta} \in \mathbb{R}^p \,:\, |\beta_1| + |\beta_2| + \ldots + |\beta_p| \leq c_1(\lambda_1)\},$$
$$\{\boldsymbol{\beta} \in \mathbb{R}^p \,:\, \beta_1^2 + \beta_2^2 + \ldots + \beta_p^2 \leq c_2(\lambda_2)\},$$

respectively, and where $c(\cdot)$ is now equipped with a subscript referring the norm to stress that it is different for lasso and ridge. The left-hand panel of Figure 4.3 visualizes these parameter constraints for $p = 2$ and $c_1(\lambda_1) = 2 = c_2(\lambda_2)$. In the Euclidean space ridge yields a spherical constraint for $\boldsymbol{\beta}$, while a diamond-like shape for the lasso. The lasso regression estimate is then that $\boldsymbol{\beta}$ inside this diamond domain which yields the smallest sum-of-squares (as is visualized by right-hand panel of Figure 4.3).

Figure 4.3: Left panel: The lasso parameter constraint ($|\beta_1| + |\beta_2| \leq 2$) and its ridge counterpart ($\beta_1^2 + \beta_2^2 \leq 2$). Solution path of the ridge estimator and its variance. Right panel: the lasso regression estimator as a constrained least squares estimtor.

The selection property of the lasso is due to the fact that the diamond-shaped parameter constraint has its corners falling on the axes. For a point to lie on an axis, one coordinate needs to equal zero. The lasso regression estimator coincides with the point inside the diamond closest to the maximum likelihood estimate. This point may correspond to a corner of the diamond, in which case one of the coordinates (regression parameters) equals zero and, consequently, the lasso regression estimator does not select this element of $\boldsymbol{\beta}$. Figure 4.4 illustrates the selection property for the case with $p = 2$ and an orthonormal design matrix. An orthornormal design matrix yields level sets (orange dotted circles in Figure 4.4) of the sum-of-squares that are spherical and centered around the maximum likelihood estimate (red dot in Figure 4.4). For maximum likelihood estimates inside the grey areas the closest point in the diamond-shaped parameter domain will be on one of its corners. Hence, for these maximum likelihood estimates the corresponding lasso regression estimate will include on a single covariate in the model. The geometrical explanation of the selection property of the lasso regression estimator also applies to non-orthonormal design matrices and in dimensions larger than two. In particular, high-dimensionally, the sum-of-squares may be a degenerated ellipsoid, that can and will still hit a corner of the diamond-shaped parameter domain. Finally, note that a zero value of lasso regression estimate does imply neither that the parameter is indeed zero nor that it will be significantly different from zero.

Larger values of the lasso penalty parameter λ_1 induce smaller parameter constraints. Consequently, the number of zero elements in the lasso regression estimator of $\boldsymbol{\beta}$ increases as λ_1 increases. However, where $\|\hat{\boldsymbol{\beta}}(\lambda_1)\|_1$ decreases monotonically as λ_1 increases (left panel of Figure 4.5 for an example and Exercise 4.4), the number of non-zero coefficients does not. Locally, at some finite λ_1, the number of non-zero elements in $\hat{\boldsymbol{\beta}}(\lambda_1)$ may increase with λ_1, to only go down again as λ_1 is sufficiently increased (as in the $\lambda_1 \rightarrow \infty$ limit the number of non-zero elements is zero, see the argumentation at the end of Section 4.2). The right panel of Figure 4.5 illustrates this behavior for an arbitrary data set.

The attractiveness of the lasso regression estimator is in its simultaneous estimation and selection of parameters. For large enough values of the penalty parameter λ_1 the estimated regression model comprises only a subset of the supplied covariates. In high-dimensions (demanding a large penalty parameter) the number of selected parameters by the lasso regression estimator is usually small (relative
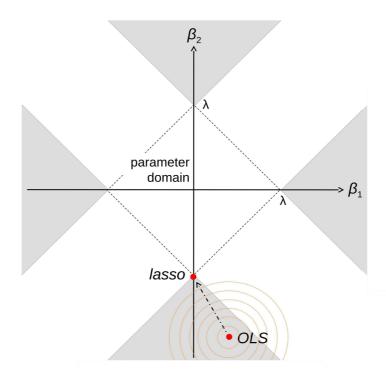
Figure 4.4: Shrinkage with the lasso. The range of possible lasso estimates is demarcated by the diamond around the origin. The grey areas contain all points that are closest to one of the diamond's corners than to any other point inside the diamond. If the OLS estimate falls inside any of these grey areas, the lasso shrinks it to the closest diamond tip (which corresponds to a sparse solution). For example, let the red dot in the fourth quadrant be an OLS estimate. It is in a grey area. Hence, its lasso estimate is the red dot at the lowest tip of the diamond.
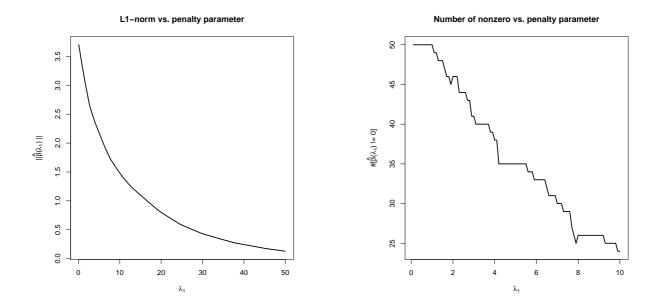

Figure 4.5: Contour plots of the sum-of-squares and the lasso regression loss (left and right panel, respectively). The dotted grey line represent level sets. The red line and dot represent the the location of minimum in both panels.

to the total number of parameters), thus producing a so-called sparse model. Would one adhere to the parsimony principle, such a sparse and thus simpler model is preferable over a full model. Simpler may be better, but too simple is worse. The phenomenon or system that is to be described by the model need

not be sparse. For instance, in molecular biology the regulatory network of the cell is no longer believed to be sparse (Boyle *et al.*, 2017). Similarly, when analyzing brain image data, the connectivity of the brain is not believed to be sparse.

4.3.1 Maximum number of selected covariates

The number of parameter/covariates selected by the lasso regression estimator is bounded non-trivially. The cardinality (i.e. the number of included covariates) of every lasso estimated linear regression model is smaller than or equal to $\min\{n, p\}$ (Bühlmann and Van De Geer, 2011). According to Bühlmann and Van De Geer (2011) this is obvious from the analysis of the LARS algorithm of Efron *et al.* (2004) (which is to be discussed in Section **??**). For now we just provide an R-script that generates the regularization paths using the `lars`-package for the `diabetes` data included in the package for a random number of samples n not exceeding the number of covariates p.

Listing 4.2 R code

```
# activate library
library(lars)

# load data
data(diabetes)
X <- diabetes$x
Y <- diabetes$y

# set sample size
n  <- sample(1:ncol(X), 1)
id <- sample(1:length(Y), n)

# plot regularization paths
plot(lars(X[id,], Y[id], intercept=FALSE))
```

Irrespective of the drawn n the plotted regularization paths all terminate before the $n + 1$-th variate enters the model. This could of course be circumstantial evidence at best, or even be labelled a bug in the software.

But even without the LARS algorithm the nontrivial part of the inequality, that the number of selected variates p does not exceed the sample size n, can be proven (Osborne *et al.*, 2000).

Theorem 4.2 (Theorem 6, Osborne *et al.*, 2000)
If $p > n$ and $\hat{\boldsymbol{\beta}}(\lambda_1)$ is a minimizer of the lasso regresssion loss function (4.2), then $\hat{\boldsymbol{\beta}}(\lambda_1)$ has at most n non-zero entries.

Proof Confer Osborne *et al.* (2000). ∎

In the high-dimensional setting, when p is large compared to n small, this implies a considerable dimension reduction. It is, however, somewhat unsatisfactory that it is the study design, i.e. the inclusion of the number of samples, that determines the upperbound of model size.

4.4 Estimation

In the absence of an analytic expression for the optimum of the lasso loss function (4.2), much attention is devoted to numerical procedures to find it.

4.4.1 Quadratic programming

In the original lasso paper Tibshirani (1996) reformulates the lasso optimization problem to a quadratic program. A quadratic problem optimizes a quadratic form subject to linear constraints. This is a well-studied optimization problem for which many readily available implementations exist (e.g., the `quadprog`-package in R). The quadratic program that is equivalent to the lasso regression problem (which minimizes the least squares criterion, $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ subject to $\|\boldsymbol{\beta}\|_1 < c(\lambda_1)$) is:

$$\min_{\mathbf{R}\boldsymbol{\beta}\geq\mathbf{0}} \tfrac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}), \tag{4.3}$$

where $\mathbf{R}$ is a $q \times p$ dimensional linear constraint matrix that specifies the linear constraints on the parameter $\boldsymbol{\beta}$. For $p = 2$ the domain implied by lasso parameter constraint $\{\boldsymbol{\beta} \in \mathbb{R}^2 : \|\boldsymbol{\beta}\|_1 < c(\lambda_1)\}$ is equal to:

$$\{\boldsymbol{\beta} \in \mathbb{R}^2 : \beta_1 + \beta_2 \le c(\lambda_1)\} \cap \{\boldsymbol{\beta} \in \mathbb{R}^2 : \beta_1 - \beta_2 \ge -c(\lambda_1)\} \cap \{\boldsymbol{\beta} \in \mathbb{R}^2 : \beta_1 - \beta_2 \le c(\lambda_1)\}$$
$$\cap \{\boldsymbol{\beta} \in \mathbb{R}^2 : \beta_1 + \beta_2 \ge -c(\lambda_1)\}.$$

This collection of linear parameter constraints can be aggregated, when using:

$$\mathbf{R} = \begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix},$$

into $\{\boldsymbol{\beta} \in \mathbb{R}^2 : \mathbf{R}\boldsymbol{\beta} \ge -c(\lambda_1)\}$.

To solve the quadratic program (4.3) it is usually reformulated in terms of its dual. Hereto we introduce the Lagrangian:

$$L(\boldsymbol{\beta}, \boldsymbol{\nu}) = \tfrac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\nu}^\top \mathbf{R}\boldsymbol{\beta}, \qquad (4.4)$$

where $\boldsymbol{\nu} = (\nu_1, \dots, \nu_q)^\top$ is the vector of non-negative multipliers. The dual function is now defined as $\inf_{\boldsymbol{\beta}} L(\boldsymbol{\beta}, \boldsymbol{\nu})$. This infimum is attained at:

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} + (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{R}^\top \boldsymbol{\nu}, \qquad (4.5)$$

which can be verified by equating the first order partial derivative with respect to $\boldsymbol{\beta}$ of the Lagrangian to zero and solving for $\boldsymbol{\beta}$. Substitution of $\boldsymbol{\beta} = \boldsymbol{\beta}^*$ into the dual function gives, after changing the minus sign:

$$\tfrac{1}{2}\boldsymbol{\nu}^\top \mathbf{R}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{R}^\top \boldsymbol{\nu} + \boldsymbol{\nu}^\top \mathbf{R}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} + \tfrac{1}{2}\mathbf{Y}^\top \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}.$$

The dual problem minimizes this expression (from which the last term is dropped as is does not involve $\boldsymbol{\nu}$) with respect to $\boldsymbol{\nu}$, subject to $\boldsymbol{\nu} \ge \mathbf{0}$. Although also a quadratic programming problem, the dual problem a) has simpler constraints and b) is defined on a lower dimensional space (if the number of columns of $\mathbf{R}$ exceeds its number of rows) than the primal problem. If $\tilde{\boldsymbol{\nu}}$ is the solution of the dual problem, the solution of the primal problem is obtained from Equation (4.5). Note that in the first term on the right hand side of Equation (4.5) we recognize the unconstrained least squares estimator of $\boldsymbol{\beta}$. Refer to, e.g., Bertsekas (2014) for more on quadratic programming.

Example 4.5 *(Orthogonal design matrix, continued)*
The evaluation of the lasso regression estimator by means of quadratic programming is illustrated using the data from the numerical Example 4.5. The R-script below solves, the implementation of the quadprog-package, the quadratic program associated with the lasso regression problem of the aforementioned example.

Listing 4.3 R code

```R
# load library
library(quadprog)

# data
Y <- matrix(c(-4.9, -0.8, -8.9, 4.9, 1.1, -2.0), ncol=1)
X <- t(matrix(c(1, -1, 3, -3, 1, 1, -3, -3, -1, 0, 3, 0), nrow=2, byrow=TRUE))

# constraint radius
L1norm <- 1.881818 + 0.8678572

# solve the quadratic program
solve.QP(t(X) %*% X, t(X) %*% Y,
         t(matrix(c(1, 1, -1, -1, 1, -1, -1, 1), ncol=2, byrow=TRUE)),
         L1norm*c(-1, -1, -1, -1))$solution
```

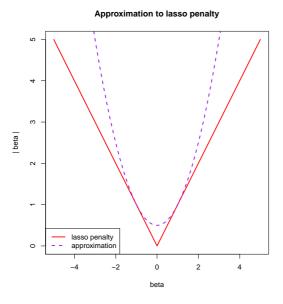The resulting estimates coincide with those found earlier. ☐

For relatively small p quadratic programming is a viable option to find the lasso regression estimator. For large p it is practically not feasible. Above the linear constraint matrix $\mathbf{R}$ is 4×2 dimensional for $p = 2$. When $p = 3$, it requires a linear constraint matrix $\mathbf{R}$ with eight rows. In general, 2^p linear constraints are required to fully specify the lasso parameter constraint on the regression parameter. Already when $p = 100$, the specification of only the linear constraint matrix $\mathbf{R}$ will take endlessly, leave alone solving the corresponding quadratic program.

4.4.2 Iterative ridge

Why develop something new, when one can also make do with existing tools? The loss function of the lasso regression estimator can be optimized by iterative application of ridge regression (as pointed out in Fan and Li, 2001). It requires an approximation of the lasso penalty, or the absolute value function. Set $p = 1$ and let β_0 be an initial parameter value for β around which the absolute value function $|\beta|$ is to be approximated. Its quadratic approximation then is:

$$|\beta| \quad \approx \quad |\beta_0| + \frac{1}{2|\beta_0^2|}(\beta^2 - \beta_0^2).$$

An illustration of this approximation is provided in the left panel of Figure 4.6.
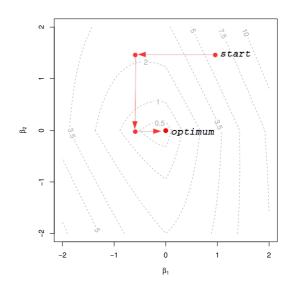

Figure 4.6: Left panel: quadratic approximation (i.e. the ridge penalty) to the absolute value function (i.e. the lasso penalty). Right panel: Illustration of the coordinate descent algorithm. The dashed grey lines are the level sets of the lasso regression loss function. The red arrows depict the parameter updates. These arrows are parallel to either the β_1 or the β_2 parameter axis, thus indicating that the regression parameter $\boldsymbol{\beta}$ is updated coordinate-wise.

The lasso regression estimator is evaluated through iterative application of the ridge regression estimator. This iterative procedure needs initiation by some guess $\boldsymbol{\beta}^{(0)}$ for $\boldsymbol{\beta}$. For example, the ridge estimator itself may serve as such. Then, at the $k + 1$-th iteration an update $\boldsymbol{\beta}^{(k+1)}$ of the lasso regression estimator of $\boldsymbol{\beta}$ is to be found. Application of the quadratic approximation to the absolute value functions of the elements of $\boldsymbol{\beta}$ (around the k-th update $\boldsymbol{\beta}^{(k)}$) in the lasso penalty yields an approximation

to the lasso regression loss function:

$$
\begin{aligned}
\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^{(k+1)}\|_2^2 + \lambda_1\|\boldsymbol{\beta}^{(k+1)}\|_1 \quad &\approx\quad \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^{(k+1)}\|_2^2 + \lambda_1\|\boldsymbol{\beta}^{(k)}\|_1 \\
&\quad + \frac{\lambda_1}{2}\sum_{j=1}^{p}\frac{1}{|\beta_j^{(k)}|}[\beta_j^{(k+1)}]^2 - \frac{\lambda_1}{2}\sum_{j=1}^{p}\frac{1}{|\beta_j^{(k)}|}[\beta_j^{(k)}]^2 \\
&\propto\quad \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^{(k+1)}\|_2^2 + \frac{\lambda_1}{2}\sum_{j=1}^{p}\frac{1}{|\beta_j^{(k)}|}[\beta_j^{(k+1)}]^2.
\end{aligned}
$$

The loss function now contains a weighted ridge penalty. In this one recognizes a generalized ridge regression loss function (see Chapter 2). As its minimizer is known, the approximated lasso regression loss function is optimized by:

$$
\boldsymbol{\beta}^{(k+1)}(\lambda_1) \quad = \quad \{\mathbf{X}^\top\mathbf{X} + \lambda_1\boldsymbol{\Psi}[\boldsymbol{\beta}^{(k)}(\lambda_1)]\}^{-1}\mathbf{X}^\top\mathbf{Y}
$$

where

$$
\text{diag}\{\boldsymbol{\Psi}[\boldsymbol{\beta}^{(k)}(\lambda_1)]\} \quad = \quad (1/|\beta_1^{(k)}|, 1/|\beta_2^{(k)}|, \ldots, 1/|\beta_p^{(k)}|).
$$

The thus generated sequence of updates $\{\boldsymbol{\beta}^{(k)}(\lambda_1)\}_{k=0}^{\infty}$ converges (under 'nice' conditions) to the lasso regression estimator $\hat{\boldsymbol{\beta}}(\lambda_1)$.

A note of caution. The in-built variable selection property of the lasso regression estimator may – for large enough choices of the penalty parameter λ_1 – cause elements of $\boldsymbol{\beta}^{(k)}(\lambda_1)$ to become arbitrary close to zero (or, in R exceed machine precision and thereby being effectively zero) after enough updates. Consequently, the ridge penalty parameter for the j-th element of regression parameter may approach infinity, as the j-th element of $\boldsymbol{\Psi}[\boldsymbol{\beta}^{(k)}(\lambda_1)]$ equals $|\beta_j^{(k)}|^{-1}$. To accommodate this, the iterative ridge regression algorithm for the evaluation of the lasso regression estimator requires a modification. Effectively, that amounts to the removal of j-th covariate from the model all together (for its estimated regression coefficient is indistinguishable from zero). After its removal, it does not return to the set of covariates. This may be problematic if two covariates are (close to) super-collinear.

4.4.3 Gradient ascent

Another method of finding the lasso regression estimator and implemented in `penalized`-package (Goeman, 2010) makes use of gradient ascent. Gradient ascent/descent is an maximization/minization method that finds the optimum of a smooth function by iteratively updating a first-order local approximation to this function. Gradient ascents runs through the following sequence of steps repetitively until convergence:
  ○ Choose a starting value.
  ○ Calculate the derivative of the function, and determine the direction in which the function increases most. This direction is the path of steepest ascent.
  ○ Proceed in this direction, until the function no longer increases.
  ○ Recalculate at this point the gradient to determine a new path of steepest ascent.
  ○ Repeat the above until the (region around the) optimum is found.
The procedure above is illustrated in Figure 4.7. The top panel shows the choice of the initial value. From this point the path of the steepest ascent is followed until the function no longer increases (right panel of Figure 4.7). Here the path of steepest ascent is updated along which the search for the optimum is proceeded (bottom panel of Figure 4.7).

The use of gradient ascent to find the lasso regression estimator is frustrated by the non-differentiability (with respect to any of the regression parameters) of the lasso penalty function at zero. In Goeman (2010) this is overcome by the use of a generalized derivative. Define the *directional* or *Gâteaux* derivative of the function $f : \mathbb{R}^p \to \mathbb{R}$ at $\mathbf{x} \in \mathbb{R}^p$ in the direction of $\mathbf{v} \in \mathbb{R}^p$ as:

$$
f'(\mathbf{x}) \quad = \quad \lim_{\tau\downarrow0}\frac{1}{\tau}\big[f(\mathbf{x}+\tau\mathbf{v}) - f(\mathbf{x})\big],
$$

assuming this limit exists. The Gâteaux derivative thus gives the infinitesimal change in f at $\mathbf{x}$ in the direction of $\mathbf{v}$. As such $f'(\mathbf{x})$ is a scalar (as is immediate from the definition when noting that $f(\cdot) \in \mathbb{R}$) and should not be confused with the gradient (the vector of partial derivatives). Furthermore, at each
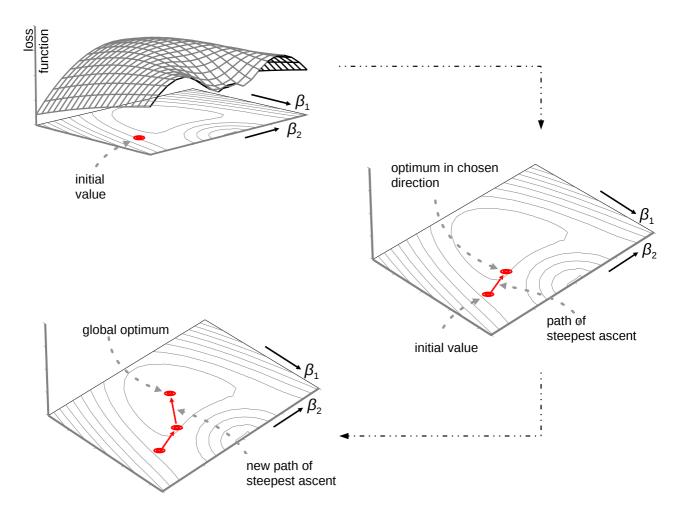
Figure 4.7: Illustration of the gradient ascent procedure.

point $\mathbf{x}$ there are infinitely many Gâteaux differentials (as there are infinitely many choices for $\mathbf{v} \in \mathbb{R}^p$). In the particular case when $\mathbf{v} = \mathbf{e}_j$, $\mathbf{e}_j$ the unit vector along the axis of the j-th coordinate, the directional derivative coincides with the partial derivative of f in the direction of x_j. Relevant for the case at hand is the absolute value function $f(x) = |x|$ with $x \in \mathbb{R}$. Evaluation of the limits in its Gâteaux derivative yields:

$$ f'(x) \;=\; \begin{cases} \mathrm{v}\frac{x}{|x|} & \text{if } x \neq 0, \\ \mathrm{v} & \text{if } x = 0, \end{cases} $$

for any $\mathrm{v} \in \mathbb{R} \setminus \{0\}$. Hence, the Gâteaux derivative of $|x|$ does exits at $x = 0$. In general, the Gâteaux differential may be uniquely defined by limiting the directional vectors $\mathbf{v}$ to $i)$ those with unit length (i.e. $\|\mathbf{v}\| = 1$) and $ii)$ the direction of steepest ascent. Using the Gâteaux derivative a gradient of $f(\cdot)$ at $\mathbf{x} \in \mathbb{R}^p$ may then be defined as:

$$ \nabla f(\mathbf{x}) \;=\; \begin{cases} f'(\mathbf{x}) \cdot \mathbf{v}_{\mathrm{opt}} & \text{if } f'(\mathbf{x}) \geq 0 \\ \mathbf{0}_p & \text{if } f'(\mathbf{x}) < 0, \end{cases} \tag{4.6} $$

in which $\mathbf{v}_{\mathrm{opt}} = \arg\max_{\{\mathbf{v}\,:\,\|\mathbf{v}\|=1\}} f'(\mathbf{x})$. This is the direction of steepest ascent, $\mathbf{v}_{\mathrm{opt}}$, scaled by Gâteaux derivative, $f'(\mathbf{x})$, in the direction of $\mathbf{v}_{\mathrm{opt}}$. Application of definition (4.7) of the Gâteaux gradient to the lasso penalized likelihood (4.2) gives:

$$ \nabla_{\mathbf{v}_{\mathrm{opt}}} \mathcal{L}_{\mathrm{lasso}}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) \;=\; \begin{cases} \mathcal{L}'_{\mathrm{lasso}}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) \cdot \mathbf{v}_{\mathrm{opt}} & \text{if } \mathcal{L}'_{\mathrm{lasso}}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) \geq 0, \\ \mathbf{0}_p & \text{if } \mathcal{L}'_{\mathrm{lasso}}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) < 0. \end{cases} \tag{4.7} $$

According to Goeman (2010) the elements of this gradient can be calculated from those of the unpenalized

log-likelihood gradient and the Gateaux derivative of the absolute value function, which yields:

$$\frac{\partial}{\partial \beta_j} \mathcal{L}_{\text{lasso}}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) = \begin{cases} \frac{\partial}{\partial \beta_j} \mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) - \lambda_1 \text{sign}(\beta_j) & \text{if} \quad \beta_j \neq 0 \\ \frac{\partial}{\partial \beta_j} \mathcal{L}(\mathbf{Y}, \mathbf{X}; \boldsymbol{\beta}) - \lambda_1 \text{sign}\left[\frac{\partial}{\partial \beta_j} \mathcal{L}(Y, X; \boldsymbol{\beta})\right] & \text{if} \quad \beta_j = 0 \text{ and } \left|\partial \mathcal{L}/\partial \beta_j\right| > \lambda_1 \\ 0 & \text{otherwise} \end{cases},$$

where $\partial \mathcal{L}/\partial \beta_j = \sum_{j'=1}^p (\mathbf{X}^\top \mathbf{X})_{j',j} \beta_j - (\mathbf{X}^\top \mathbf{Y})_j$.

Convergence of gradient ascent can be slow close to the optimum. This is due to its linear approximation of the function. Close to the optimum the linear term of the Taylor expansion vanishes and is dominated by the second-order quadratic term. To speed-up convergence close to the optimum the gradient ascent implementation offered by the `penalized`-package switches to a Newton-Raphson procedure.

4.4.4 Coordinate descent

Coordinate descent is another optimization algorithm that may be used to evaluate the lasso regression estimator numerically, as is done by the implemention offered via the `glmnet`-package. Coordinate descent, instead of following the gradient of steepest descent (as in Section 4.4.3), minimizes the loss function along the coordinates one-at-the-time. For the j-th regression parameter this amounts to finding:

$$\begin{aligned} \arg\min_{\beta_j} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 &= \arg\min_{\beta_j} \|\mathbf{Y} - \mathbf{X}_{*,\backslash j}\boldsymbol{\beta}_{\backslash j} - \mathbf{X}_{*,j}\boldsymbol{\beta}_j\|_2^2 + \lambda|\boldsymbol{\beta}_j|_1 \\ &= \arg\min_{\beta_j} \|\tilde{\mathbf{Y}} - \mathbf{X}_{*,j}\boldsymbol{\beta}_j\|_2^2 + \lambda|\boldsymbol{\beta}_j|_1, \end{aligned}$$

where $\tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}_{*,\backslash j}\boldsymbol{\beta}_{\backslash j}$. After a simple rescaling of both $\mathbf{X}_{*,j}$ and β_j, the minimization of the lasso regression loss function with respect to β_j is equivalent to one with an orthonormal design matrix. From Example 4.4 it is known the minimizer is obtained by application of the soft-threshold function to the corresponding maximum likelihood estimator (now derived from $\tilde{\mathbf{Y}}$ and $\mathbf{X}_j$). The coordinate descent algorithm iteratively runs over the p elements until convergence. The right panel of Figure 4.6 provides an illustration of the coordinate descent algorithm.

Convergence of the coordinate descent algorithm to the minimum of the lasso regression loss function (4.2) is warranted by the convexity of this function. At each minization step the coordinate descent algorithm yields an update of the parameter estimate that corresponds to an equal or smaller value of the loss function. It, together with the compactness of diamond-shaped parameter domain and the boundedness (from below) of the lasso regression loss function, implies that the coordinate descent algorithm converges to the minimum of this lasso regression loss function.

4.5 Moments

In general the moments of the lasso regression estimator appear to be unknown. In certain cases an approximation may be given. This is pointed out here. Use the quadratic approximation to the absolute value function of Section 4.4.2 and approximate the lasso regression loss function around the lasso regression estimate:

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \quad \approx \quad \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \frac{\lambda_1}{2}\sum_{j=1}^p \frac{1}{|\hat{\beta}(\lambda_1)|}\beta_j^2.$$

Optimization of the right-hand side of the preceeding display with respect to $\boldsymbol{\beta}$ gives a 'ridge approximation' to the lasso estimate:

$$\hat{\boldsymbol{\beta}}(\lambda_1) \quad \approx \quad \{\mathbf{X}^\top \mathbf{X} + \lambda_1 \boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1} \mathbf{X}^\top \mathbf{Y},$$

with $(\boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)])_{jj} = |\hat{\beta}_j(\lambda_1)|^{-1}$ if $\hat{\beta}_j(\lambda_1) \neq 0$. Now use this 'ridge approximation' to obtain the approximation to the moments of the lasso regression estimator:

$$\begin{aligned} \mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda_1)] &\approx \mathbb{E}\left(\{\mathbf{X}^\top \mathbf{X} + \lambda_1 \boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1} \mathbf{X}^\top \mathbf{Y}\right) \\ &= \{\mathbf{X}^\top \mathbf{X} + \lambda_1 \boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1} \mathbf{X}^\top \mathbf{X} \mathbb{E}[(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}] \\ &= \{\mathbf{X}^\top \mathbf{X} + \lambda_1 \boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1} \mathbf{X}^\top \mathbf{X} \boldsymbol{\beta} \end{aligned}$$

and

$$\begin{aligned}
\mathrm{Var}[\hat{\boldsymbol{\beta}}(\lambda_1)] &\approx \mathrm{Var}\big(\{\mathbf{X}^\top\mathbf{X} + \lambda_1\boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1}\mathbf{X}^\top\mathbf{Y}\big) \\
&= \{\mathbf{X}^\top\mathbf{X} + \lambda_1\boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1}\mathbf{X}^\top\mathbf{X}\mathrm{Var}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}]\mathbf{X}^\top\mathbf{X}\{\mathbf{X}^\top\mathbf{X} + \lambda_1\boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1} \\
&= \sigma^2\{\mathbf{X}^\top\mathbf{X} + \lambda_1\boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1}\mathbf{X}^\top\mathbf{X}\{\mathbf{X}^\top\mathbf{X} + \lambda_1\boldsymbol{\Psi}[\hat{\boldsymbol{\beta}}(\lambda_1)]\}^{-1}.
\end{aligned}$$

These approximations can only be used when the lasso regression estimate is not sparse, which is at odds with its attractiveness. A better approximation of the variance of the lasso regression estimator can be found in Osborne *et al.* (2000), but even this becomes poor when many elements of $\boldsymbol{\beta}$ are estimated as zero.

Even if these approximations are only crude, they indicate that the moments of the lasso regression estimator exhibit similar behaviour as those of its ridge counterpart. The (approximation of the) mean $\mathbb{E}[\hat{\boldsymbol{\beta}}(\lambda_1)]$ tends to zero as $\lambda_1 \to \infty$. This was intuitively already expected from the form of the lasso regression loss function (4.2), in which the penalty term dominates for large λ_1 and is minimized for $\hat{\boldsymbol{\beta}}(\lambda_1) = \mathbf{0}_p$. This may also be understood geometrically when appealing to the equivalent constrained estimation formation of the lasso regression estimator. The parameter constraint shrinks to zero with increasing λ_1. Hence, so must the estimator. Similarly, the (approximation of the) variance of the lasso regression estimator vanishes as the penalty parameter λ_1 grows. Again, its loss function (4.2) provides the intuition: for large λ_1 the penalty term, which does not depend on data, dominates. Or, from the perspective of the constrained estimation formulation, the parameter constraint shrinks to zero as $\lambda_1 \to \infty$. Hence, so must the variance of the estimator, as less and less room is left for it to fluctuate.

The behaviour of the mean squared error, bias squared plus variance, of the lasso regression estimator in terms of λ_1 is hard to characterize exactly without knowledge of the quality of the approximations. In particular, does a λ_1 exists such that the MSE of the lasso regression estimator outperforms that of its maximum likelihood counterpart. Nonetheless, a first observation may be obtained from reasoning in extremis. Suppose $\boldsymbol{\beta} = \mathbf{0}_p$, which corresponds to an empty or maximally sparse model. A large value of λ_1 then yields a zero estimate of the regression parameter: $\hat{\boldsymbol{\beta}}(\lambda_1) = \mathbf{0}_p$. The bias squared is thus minimized: $\|\hat{\boldsymbol{\beta}}(\lambda_1) - \boldsymbol{\beta}\|_2^2 = 0$. With the bias vanished and the (approximation of the) variance decreasing in λ_1, so must the MSE decrease for λ_1 larger than some value. So, for an empty model the lasso regression estimator with a sufficiently large penalty parameter yields a better MSE than the maximum likelihood estimator. For very sparse models this property may be expected to uphold, but for non-sparse models the bias squared will have a substantial contribution to the MSE, and it is thus not obvious whether a λ_1 exists that yields a favourable MSE for the lasso regression estimator. This is investigated *in silico* in Hansen (2015). The simulations presented there indicate that the MSE of the lasso regression estimator is particularly sensitive to the actual $\boldsymbol{\beta}$. Moreover, for a large part of the parameter space $\boldsymbol{\beta} \in \mathbb{R}^p$ the MSE of $\hat{\boldsymbol{\beta}}(\lambda_1)$ is behind that of the maximum likelihood estimator.

4.6 The Bayesian connection

The lasso regression estimator, being a penalized estimator, knows a Bayesian formulation, much like the (generalized) ridge regression estimator could be viewed as a Bayesian estimator when imposing a Gaussian prior (cf. Sections 1.6 and 2.2). Instead of normal prior, the lasso regression estimator requires (as suggested by the form of the lasso penalty) a zero-centered Laplacian (or double exponential) prior to be viewed as a Bayesian estimator. A zero-centered Laplace distributed random variable X has density $f_X(x) = \frac{1}{2b}\exp(-|x|/b)$ with scale parameter $b > 0$. The top panel of Figure 4.8 shows the Laplace prior, and for contrast the normal prior of the ridge regression estimator. This figure reveals that the 'lasso prior' puts more mass close to zero and in the tails than the Gaussian 'ridge prior'. This corroborates with the tendency of the lasso regression estimator to produce either zero or large (compared to ridge) estimates.

The lasso regression estimator corresponds to the maximum a posteriori (MAP) estimator of $\boldsymbol{\beta}$, when the prior is a Laplace distribution. The posterior distribution is then proportional to:

$$\prod_{i=1}^{n}(2\pi\sigma^2)^{-1/2}\exp[-(2\sigma^2)^{-1}(Y_i\mathbf{X}_{i,*}\boldsymbol{\beta})^2] \times \prod_{j=1}^{p}(2b)^{-1}\exp(-|\beta_j|/b).$$

The posterior is not a well-known and characterized distribution. This is not necessary as interest concentrates here on its maximum. The location of the posterior mode coincides with the location of the
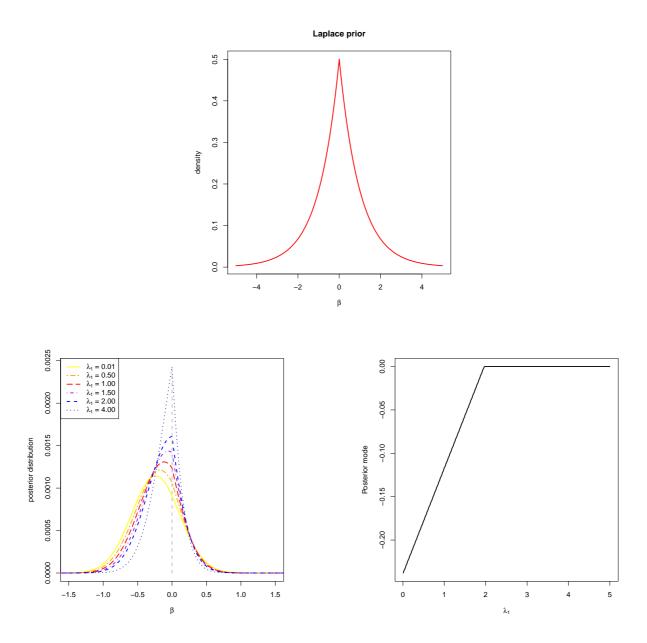
Figure 4.8: Solution path of the ridge estimator and its variance. The left panel shows the solution path of the ridge estimator for the data of Example 1.3. In the right panel the corresponding variance of the ridge estimator is plotted against the (logarithm of the) penalty parameter.

maximum of logarithm of the posterior. The log-posterior is proportional to: $-(2\sigma^2)^{-1}\|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 - b^{-1}\|\boldsymbol{\beta}\|_1$, with its maximizer minimizing $\|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + (2\sigma^2/b)\|\boldsymbol{\beta}\|_1$. In this one recognizes the form of the lasso regression loss function (4.2). It is thus clear that the scale parameter of the Laplace distribution reciprocally relates to lasso penalty parameter λ_1, similar to the relation of the ridge penalty parameter λ_2 and the variance of the Gaussian prior of the ridge regression estimator.

The posterior may not be a standard distribution, in the univariate case ($p = 1$) it is can visualized. Specifically, the behaviour of the MAP can then be illustrated, which – as the MAP estimator corresponds to the lasso regression estimator – should also exhibit the selection property. The bottom left panel of Figure 4.8 shows the posterior distribution for various choices of the Laplace scale parameter (i.e. lasso penalty parameter). Clearly, the mode shifts towards zero as the scale parameter decreases / lasso penalty parameter increases. In particular, the posterior obtained from the Laplace prior with the smallest scale parameter (i.e. largest penalty parameter λ_1), although skewed to the left, has a mode placed exactly at zero. The Laplace prior may thus produce MAP estimators that select. However, for smaller values of the

lasso penalty parameter the Laplace prior is not concentrated enough around zero and the contribution of the likelihood in the posterior outweighs that of the prior. The mode is then not located at zero and the parameter is 'selected' by the MAP estimator. The bottom right panel of Figure 4.8 plots the mode of the normal-Laplace posterior vs. the Laplace scale parameter. In line with Theorem 4.1 it is piece-wise linear.

Park and Casella (2008) go beyond the elementary correspondence of the frequentist lasso estimator and the Bayesian posterior mode and formulate the Bayesian lasso regression model. To this end they exploit the fact that the Laplace distribution can be written as a scale mixture of normal distributions with an exponentiona mixing density. This allows the construction of a Gibbs sampler for the Bayesian lasso estimator. Finally, they suggest to impose a gamma-type hyperprior on the (square of the) lasso penalty parameter. Such a full Bayesian formulation of the lasso problem enables the construction of credible sets (i.e. the Bayesian counterpart of confidence intervals) to express the uncertainty of the maximum a posterior estimator. However, the lasso regression estimator may be seen as a Bayesian estimator, in the sense that it coincides with the posterior mode, the 'lasso' posterior distribution cannot be blindly used for uncertainty quantification. In high-dimensional sparse settings the 'lasso' posterior distribution of $\boldsymbol{\beta}$ need not concentrate around the true parameter, even though its mode is a good estimator of the regression parameter (cf. Section 3 and Theorem 7 of Castillo *et al.*, 2015).
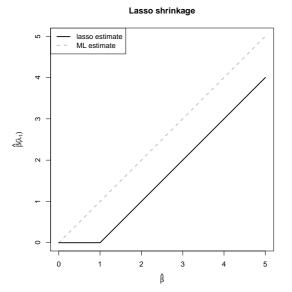
4.7 Comparison to ridge

Here an inventory of the similarities and differences between the lasso and ridge regression estimators is presented. To recap what we have seen so far: both estimators optimize a loss function of the form (4.1) and can be viewed as Bayesian estimators. But in various respects the lasso regression estimator exhibited differences from its ridge counterpart: *i)* the former need not be uniquely defined (for a given value of the penalty parameter) whereas the latter is, *ii)* an analytic form of the lasso regression estimator does in general not exists, but *iii)* it is sparse (for large enough values of the lasso penalty parameter). The remainder of this section expands this inventory.

4.7.1 Linearity

The ridge regression estimator is a linear (in the observations) estimator, while the lasso regression estimator is not. This is immediate from the analytic expression of the ridge regression estimator, $\hat{\boldsymbol{\beta}}(\lambda_2) = (\mathbf{X}^\top \mathbf{X} + \lambda_2 \mathbf{I}_{pp})^{-1} \mathbf{X}^\top \mathbf{Y}$, which is a linear combination of the observations $\mathbf{Y}$. To show the non-linearity of the lasso regression estimator available, it suffices to study the analytic expression of j-th element of $\hat{\boldsymbol{\beta}}(\lambda_1)$ in the orthonormal case: $\hat{\beta}_j(\lambda_1) = \text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \frac{1}{2}\lambda_1)_+ = \text{sign}(\mathbf{X}_{*,j}^\top \mathbf{Y})(|\mathbf{X}_{*,j}^\top \mathbf{Y}| - \frac{1}{2}\lambda_1)_+$. This clearly is not linear in $\mathbf{Y}$. Consequently, the response $\mathbf{Y}$ may be scaled by some constant c, denoted $\tilde{\mathbf{Y}} = c\mathbf{Y}$, and the corresponding ridge regression estimators are one-to-one related by this same factor $\hat{\boldsymbol{\beta}}(\lambda_2) = c\tilde{\boldsymbol{\beta}}(\lambda_2)$. The lasso regression estimator based on the unscaled data is not so easily recovered from its counterpart obtained from the scaled data.

4.7.2 Shrinkage

Both lasso and ridge regression estimation minimize the sum-of-squares plus a penalty. The latter encourages the estimator to be small, in particular closer to zero. This behavior is called shrinkage. The particular form of the penalty yields different types of this shrinkage behavior. This is best grasped in the case of an orthonormal design matrix. The j-the element of the ridge regression estimator then is: $\hat{\beta}_j(\lambda_2) = (1 + \lambda_2)\hat{\beta}_j$, while that of the lasso regression estimator is: $\hat{\beta}_j(\lambda_1) = \text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \frac{1}{2}\lambda_1)_+$. In Figure 4.9 these two estimators $\hat{\beta}_j(\lambda_2)$ and $\hat{\beta}_j(\lambda_1)$ are plotted as a function of the maximum likelihood estimator $\hat{\beta}_j$. Figure 4.9 shows that lasso and ridge regression estimator translate and scale, respectively, the maximum likelihood estimator, which could also have been concluded from the analytic expression of both estimators. The scaling of the ridge regression estimator amounts to substantial and little shrinkage (in an absolute sense) for elements of the regression parameter $\boldsymbol{\beta}$ with a large and small maximum likelihood estimate, respectively. In contrast, the lasso regression estimator applies an equal amount of shrinkage to each element of $\boldsymbol{\beta}$, irrespective of the coefficients' sizes.
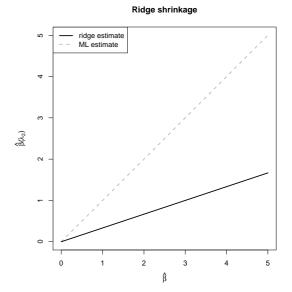
Figure 4.9: Solution path of the ridge estimator and its variance. The left panel shows the solution path of the ridge estimator for the data of Example 1.3. In the right panel the corresponding variance of the ridge estimator is plotted against the (logarithm of the) penalty parameter.

4.7.3   Simulation I: Covariate selection

Here it is investigated whether lasso regression exhibits the same behaviour as ridge regression in the presence of covariates with differing variances. Recall: the simulation of Section 1.10.1 showed that ridge regression shrinks the estimates of covariates with a large spread less than those with a small spread. That simulation has been repeated, with the exact same parameter choices and sample size, but now with the ridge regression estimator replaced by the lasso regression estimator. To refresh the memory: in the simulation of Section 1.10.1 the linear regression model is fitted, now with the lasso regression estimator. The $(n = 1000) \times (p = 50)$ dimensional design matrix $\mathbf{X}$ is sampled from a multivariate normal distribution: $\mathbf{X}_{i,*}^{\top} \sim \mathcal{N}(\mathbf{0}_{50}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ diagonal and $(\boldsymbol{\Sigma})_{jj} = j/10$ for $j = 1, \ldots, p$. The response $\mathbf{Y}$ is generated through $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\beta}$ a vector of all ones and $\boldsymbol{\varepsilon}$ sampled from the multivariate standard normal distribution. Hence, all covariates contribute equally to the response.

The results of the simulation are displayed in Figure 4.10, which shows the regularization paths of the $p = 50$ covariates. The regularization paths are demarcated by color and style to indicate the size of the spread of the corresponding covariate. These regularization paths show that the lasso regression estimator shrinks – like the ridge regression estimator – the covariates with the smallest spread most. For the lasso regression this translates (for sufficiently large values of the penalty parameter) into a preference for the selection of covariates with largest variance.

Intuition for this behavior of the lasso regression estimator may be obtained through geometrical arguments analogous to that provided for the similar behaviour of the ridge regression estimator in Section 1.10.1. Algebraically it is easily seen when assuming an orthonormal design with $\mathrm{Var}(X_1) \gg \mathrm{Var}(X_2)$. The lasso regression loss function can then be rewritten, as in Example 4.5, to:
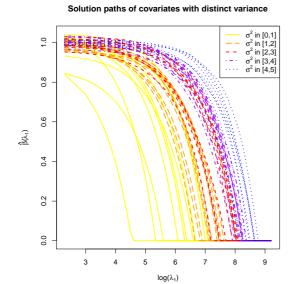
$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \quad = \quad \|\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\gamma}\|_2^2 + \lambda_1[\mathrm{Var}(X_1)]^{-1/2}|\gamma_1| + \lambda_1[\mathrm{Var}(X_2)]^{-1/2}|\gamma_2|,$$

where $\gamma_1 = [\mathrm{Var}(X_1)]^{1/2}\beta_1$ and $\gamma_2 = [\mathrm{Var}(X_2)]^{1/2}\beta_2$. The rescaled design matrix $\tilde{\mathbf{X}}$ is now orthonormal and analytic expressions of estimators of γ_1 and γ_2 are available. The former parameter is penalized substantially less than the latter as $\lambda_1[\mathrm{Var}(X_1)]^{-1/2} \ll \lambda_1[\mathrm{Var}(X_2)]^{-1/2}$. As a result, if for large enough values of λ_1 one variable is selected, it is more likely to be γ_1.

4.7.4   Simulation II: correlated covariates

The behaviour of the lasso regression estimator is now studied in the presence of collinearity among the covariates. Previously, in simulation, Section 1.10.2, the ridge regression estimator was shown to exhibit
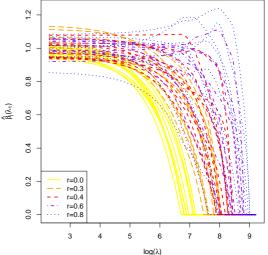
Figure 4.10: Solution path of the ridge estimator and its variance. The left panel shows the solution path of the ridge estimator for the data of Example 1.3. In the right panel the corresponding variance of the ridge estimator is plotted against the (logarithm of the) penalty parameter.

the joint shrinkage of strongly collinear covariates. This simulation is repeated for the lasso regression estimator. The details of the simulation are recapped. The linear regression model is fitted by means of the lasso regression estimator. The $(n = 1000) \times (p = 50)$ dimensional design matrix $\mathbf{X}$ is samples from a multivariate normal distribution: $\mathbf{X}_{i,*}^{\top} \sim \mathcal{N}(\mathbf{0}_{50}, \mathbf{\Sigma})$ with a block-diagonal $\mathbf{\Sigma}$. The k-the, $k = 1, \ldots, 5$, diagonal block, denoted $\mathbf{\Sigma}_{kk}$ comprises ten covariates and equals $\frac{k-1}{5} \mathbf{1}_{10 \times 10} + \frac{6-k}{5} \mathbf{I}_{10 \times 10}$ for $k = 1, \ldots, 5$. The response vector $\mathbf{Y}$ is then generated by $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon}$ sampled from the multivariate standard normal distribution and $\boldsymbol{\beta}$ containing only ones. Again, all covariates contribute equally to the response.

The results of the above simulation results are captured in Figure 4.10. It shows the lasso regularization paths for all elements of the regression parameter $\boldsymbol{\beta}$. The regularization paths of covariates corresponding to the same block of $\mathbf{\Sigma}$ (indicative of the degree of collinearity) are now marcated by different colors and styles. Whereas the ridge regularization paths nicely grouped per block, the lasso counterparts do not. The selection property spoils the party. Instead of shrinking the regression parameter estimates of collinear covariates together, the lasso regression estimator (for sufficiently large values of its penalty parameter λ_1) tends to pick one covariates to enters the model while forcing the others out (by setting their estimates to zero).

4.8 Exercises

Question 4.1
Find the lasso regression solution for the data below for a general value of λ and for the straight line model $Y = \beta_0 + \beta_1 X + \varepsilon$ (only apply the lasso penalty to the slope parameter, not to the intercept). Show that when λ is chosen as 7, the lasso solution fit is $\hat{Y} = 40 + 1.75X$. Data: $\mathbf{X}^{\top} = (X_1, X_2, \ldots, X_8)^{\top} = (-2, -1, -1, -1, 0, 1, 2, 2)^T$, and $\mathbf{Y}^{\top} = (Y_1, Y_2, \ldots, Y_8)^{\top} = (35, 40, 36, 38, 40, 43, 45, 43)^{\top}$.

Question 4.2
Show the non-uniqueness of the lasso regression estimator for $p > 2$ when the design matrix $\mathbf{X}$ contains linearly dependent columns.

Question 4.3
Derive an analytic expression for the lasso regression estimator. In this assume that the columns of $\mathbf{X}$ have been standardized (i.e. have a zero mean and unit variance) and have a positive correlation ρ.

Question 4.4
Show $\|\hat{\boldsymbol{\beta}}(\lambda_1)\|_1$ is monotone increasing in λ_1. In this assume orthonormality of the design matrix $\mathbf{X}$.

Question 4.5
Consider the standard linear regression model $Y_i = X_{i,1}\beta_1 + X_{i,2}\beta_2 + \varepsilon_i$ for $i = 1, \ldots, n$ and with the ε_i i.i.d. normally distributed with zero mean and a common variance. In the estimation of the regression parameter $(\beta_1, \beta_2)^\top$ a lasso penalty is used: $\lambda_{1,1}|\beta_1| + \lambda_{1,2}|\beta_2|$ with penalty parameters $\lambda_{1,1}, \lambda_{1,2} > 0$.

    a) Let $\lambda_{1,1} = \lambda_{1,2}$ and assume the covariates are orthogonal with the spread of the first covariate being much larger than that of the second. Draw a plot with β_1 and β_2 on the x- and y-axis, repectively. Sketch the parameter constraint as implied by the lasso penalty. Add the levels sets of the sum-of-squares, $\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$, loss criterion. Use the plot to explain why the lasso tends to select covariates with larger spread.

    b) Assume the covariates to be orthonormal. Let $\lambda_{1,2} \gg \lambda_{1,1}$. Redraw the plot of part a of this exercise. Use the plot to explain the effect of differening $\lambda_{1,1}$ and $\lambda_{1,2}$ on the resulting lasso estimate.

    c) Show that the two cases (i.e. the assumptions on the covariates and penalty parameters) of part a and b of this exercise are equivalent, in the sense that their loss functions can be rewritten in terms of the other.

Question 4.6
Investigate the effect of the variance of the covariates on variable selection by the lasso. Hereto consider the toy model: $Y_i = X_{1i} + X_{2i} + \varepsilon_i$, where $\epsilon_i \sim \mathcal{N}(0,1)$, $X_{1i} \sim \mathcal{N}(0,1)$, and $X_{2i} = a\,X_{1i}$ with $a \in [0,2]$. Draw a hundred samples for both X_{1i} and ε_i and construct both X_{2i} and Y_i for a grid of a's. Fit the model by lasso regression with $\lambda = 1$ for each choice of a. Plot e.g. in one figure *a)* the variance of X_{i1}, *b)* the variance of X_{2i}, and *c)* the indicator of the selection of X_{2i}. Which covariate is selected for which values of a?

Question 4.7
Augment the lasso penalty with the sum of the absolute differences all pairs of successive regression coefficients:

$$\lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_F \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|.$$

This augmented lasso penalty is referred to as the *fused lasso penalty*.

    a) Consider the standard multiple linear regression model:

$$Y_i \;\; = \;\; \sum_{j=1}^{p} X_{ij}\,\beta_j + \varepsilon_i.$$

       Estimation of the regression parameters takes place via minimization of penalized sum of squares, in which the fused lasso penalty is used with $\lambda_1 = 0$. Rewrite the corresponding loss function to the standard lasso problem by application of the following change-of-variables: $\gamma_1 = \beta_1$ and $\gamma_j = \beta_j - \beta_{j-1}$.

    b) Investigate on simulated data the effect of the second summand of the fused lasso penalty on the parameter estimates. In this, temporarily set $\lambda_1 = 0$.

    c) Let λ_1 equal zero still. Compare the regression estimates of Question 4b to the ridge estimates with a first-order autoregressive prior. What is qualitatively the difference in the behavior of the two estimates? *Hint:* plot the full solution path for the penalized estimates of both estimation procedures.

    d) How do the estimates of part b of this question change if we allow $\lambda_1 > 0$?

Question 4.8
A researcher has measured gene expression measurements for 1000 genes in 40 subjects, half of them cases and the other half controls.

    a) Describe and explain what would happen if the researcher would fit an ordinary logistic regression to these data, using case/control status as the response variable.

b) Instead, the researcher chooses to fit a lasso regression, choosing the tuning parameter lambda by cross-validation. Out of 1000 genes, 37 get a non-zero regression coefficient in the lasso fit. In the ensuing publication, the researcher writes that the 963 genes with zero regression coefficients were found to be "irrelevant". What is your opinion about this statement?

Question 4.9

Consider the standard linear regression model $Y_i = \mathbf{X}_{i,*}\boldsymbol{\beta} + \varepsilon_i$ for $i = 1, \dots, n$ and with the ε_i i.i.d. normally distributed with zero mean and a common variance. Let the first covariate correspond to the intercept. The model is fitted to data by means of the minimization of the sum-of-squares augmented with a lasso penalty in which the intercept is left unpenalized: $\lambda_1 \sum_{j=2}^{p} |\beta_j|$ with penalty parameter $\lambda_1 > 0$. The penalty parameter is chosen through leave-one-out cross-validation (LOOCV). The predictive performance of the model is evaluated, again by means of LOOCV. Thus, creating a double cross-validation loop. At each inner loop the optimal λ_1 yields an empty intercept-only model, from which a prediction for the left-out sample is obtained. The vector of these prediction is compared to the corresponding observation vector through their Spearman correlation (which measures the monotonicity of a relatonship and – as a correlation measure – assumed values on the $[-1, 1]$ interval with an analogous interpretation to the 'ordinary' correlation). The latter equals -1. Why?

Question 4.10

Download the `breastCancerNKI` package from BioConductor:
```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite("breastCancerNKI")
```
Activate the library and load leukemia data from the package:
```
> library(breastCancerNKI)
> data(nki)
```
The eset-object `nki` is now available. It contains the expression profiles of 337 breast cancer patients. Each profile comprises expression levels of 24481 genes. Extract the expression data from the object, remove all genes with missing values, center the gene expression gene-wise around zero, and limit the data set to the first thousand genes. The reduction of the gene dimensionality is only for computational speed.
```
X <- exprs(nki)
X <- X[-which(rowSums(is.na(X)) > 0),]
X <- apply(X[1:1000,], 1, function(X) X - mean(X) ) .
```
Furthermore, extract the estrogen receptor status (short: ER status), an important prognostic indicator for breast cancer.
```
Y <- pData(nki)[,8]
```
a) Relate the ER status and the gene expression levels by a logistic regression model, which is fitted by means of ridge penalized maximum likelihood. First, find the optimal value of the penalty parameter of λ by means of cross-validation. This is implemented in `optL2`-function of the `penalized`-package available from `CRAN`.
b) Evaluate whether the cross-validated likelihood indeed attains a maximum at the optimal value of λ. This can be done with the `profL2`-function of the `penalized`-package available from `CRAN`.
c) Investigate the sensitivity of the penalty parameter selection with respect to the choice of the cross-validation fold.
d) Does the optimal lambda produce a reasonable fit? And how does it compare to the 'ridge fit'?

Question 4.11

Consider fitting a multiple linear regression model by means of elastic net penalized least squares.
a) Recall the data augmentation trick of Question of the ridge regression exercises. Use the same trick to show that the elastic net least squares loss function can be reformulated to the form of the traditional lasso function. *Hint*: absorb the ridge part of the elastic net penalty into the sum of squares.
b) The lasso can select maximally $\min\{n, p\} = rank(\mathbf{X})$ covariates. How many covariates can – in principle – the elastic net select?

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**(6), 716–723.

Allen, D. M. (1974). The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, **16**(1), 125–127.

Ambs, S., Prueitt, R. L., Yi, M., Hudson, R. S., Howe, T. M., Petrocca, F., Wallace, T. A., Liu, C.-G., Volinia, S., Calin, G. A., Yfantis, H. G., Stephens, R. M., and Croce, C. M. (2008). Genomic profiling of microrna and messenger RNA reveals deregulated microrna expression in prostate cancer. *Cancer Research*, **68**(15), 6162–6170.

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis (3rd edition)*. John Wiley & Sons.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, **9**, 485–76.

Bartel, D. P. (2004). MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, **116**(2), 281–297.

Bertsekas, D. P. (2014). *Constrained Optimization and Lagrange Multiplier Methods*. Academic press.

Bickel, P. J. and Doksum, K. A. (2001). *Mathematical Statistics, Vol. I*. Prentice Hall, Upper Saddle River, New Jersey.

Bijma, F., Jonker, M. A., and van der Vaart, A. W. (2017). *An introduction to mathematical statistics*. Amsterdam University Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Boyle, E. A., Li, Y. I., and Pritchard, J. K. (2017). An expanded view of complex traits: from polygenic to omnigenic. *Cell*, **169**(7), 1177–1186.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media.

Cancer Genome Atlas Network (2011). Integrated genomic analyses of ovarian carcinoma. *Nature*, **474**(7353), 609–615.

Cancer Genome Atlas Network (2012). Comprehensive molecular characterization of human colon and rectal cancer. *Nature*, **487**(7407), 330–337.

Castillo, I., Schmidt-Hieber, J., and Van der Vaart, A. W. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics*, **43**(5), 1986–2018.

Draper, N. R. and Smith, H. (1998). *Applied Regression Analysis (3rd edition)*. John Wiley & Sons.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, **32**(2), 407–499.

Esquela-Kerscher, A. and Slack, F. J. (2006). Oncomirs: microRNAs with a role in cancer. *Nature Reviews Cancer*, **6**(4), 259–269.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, **96**(456), 1348–1360.

Farebrother, R. W. (1976). Further results on the mean square error of ridge regression. *Journal of the Royal Statistical Society, Series B (Methodological)*, pages 248–250.

Fletcher, R. (2008). *Practical Methods of Optimization, 2nd Edition*. John Wiley, New York.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, **1**(2), 302–332.

Goeman, J. J. (2010). L_1 penalized estimation in the Cox proportional hazards model. *Biometrical Journal*, **52**, 70–84.

Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **21**(2), 215–223.

Hansen, B. E. (2015). The risk of James–Stein and lasso shrinkage. *Econometric Reviews*, **35**(8-10), 1456–1470.

Harville, D. A. (2008). *Matrix Algebra From a Statistician's Perspective*. Springer, New York.

Hastie, T. and Tibshirani, R. (2004). Efficient quadratic regularization for expression arrays. *Biostatistics*, **5**(3), 329–340.

Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The Elements of Statistical Learning*. Springer.

Hemmerle, W. J. (1975). An explicit solution for generalized ridge regression. *Technometrics*, **17**(3), 309–314.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**(1), 55–67.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied Logistic Regression*, volume 398. John Wiley & Sons.

Kim, V. N. and Nam, J.-W. (2006). Genomics of microRNA. *TRENDS in Genetics*, **22**(3), 165–173.

Lawless, J. F. (1981). Mean squared error properties of generalized ridge estimators. *Journal of the American Statistical Association*, **76**(374), 462–466.

Le Cessie, S. and Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied Statistics*, **41**(1), 191–201.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, **88**, 365–411.

Leeb, H. and Pötscher, B. M. (2008). Sparse estimators and the oracle property, or the return of Hodges' estimator. *Journal of Econometrics*, **142**(1), 201–211.

Leng, C., Lin, Y., and Wahba, G. (2006). A note on the lasso and related procedures in model selection. *Statistica Sinica*, pages 1273–1284.

Mathai, A. M. and Provost, S. B. (1992). *Quadratic Forms in Random Variables: Theory and Applications*. Dekker.

Meijer, R. J. and Goeman, J. J. (2013). Efficient approximate k-fold and leave-one-out cross-validation for ridge regression. *Biometrical Journal*, **55**(2), 141–155.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(4), 417–473.

Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). On the lasso and its dual. *Journal of Computational and Graphical Statistics*, **9**(2), 319–337.

Padmanabhan, V., Callas, P., Philips, G., Trainer, T., and Beatty, B. (2004). DNA replication regulation protein MCM7 as a marker of proliferation in prostate cancer. *Journal of Clinical Pathology*, **57**(10), 1057–1062.

Park, T. and Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association*, **103**(482), 681–686.

Pust, S., Klokk, T., Musa, N., Jenstad, M., Risberg, B., Erikstein, B., Tcatchoff, L., Liestøl, K., Danielsen, H., Van Deurs, B., and K, S. (2013). Flotillins as regulators of ErbB2 levels in breast cancer. *Oncogene*, **32**(29), 3443–3451.

Rao, C. R. (1973). *Linear Statistical Inference and its Applications*. John Wiley & Sons.

Rosset, S. and Zhu, J. (2007). Piecewise linear regularized solution paths. *The Annals of Statistics*, pages 1012–1030.

Sardy, S. (2008). On the practice of rescaling covariates. *International Statistical Review*, **76**(2), 285–297.

Schaefer, R. L., Roi, L. D., and Wolfe, R. A. (1984). A ridge logistic estimator. *Communications in Statistics: Theory and Methods*, **13**(1), 99–113.

Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Application in Genetics and Molecular Biology*, **4**, Article 32.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**(2), 461–464.

Sterner, J. M., Dew-Knight, S., Musahl, C., Kornbluth, S., and Horowitz, J. M. (1998). Negative regulation of DNA replication by the retinoblastoma protein is mediated by its association with MCM7. *Molecular and Cellular Biology*, **18**(5), 2748–2757.

Subramanian, J. and Simon, R. (2010). Gene expression–based prognostic signatures in lung cancer: ready for clinical use? *Journal of the National Cancer Institute*, **102**(7), 464–474.

Theobald, C. M. (1974). Generalizations of mean square error applied to ridge regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, **36**(1), 103–106.

Tibshirani, R. (1996). Regularized shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, **58**(1), 267–288.

Tibshirani, R. J. (2013). The lasso problem and uniqueness. *Electronic Journal of Statistics*, **7**, 1456–1490.

Tye, B. K. (1999). MCM proteins in DNA replication. *Annual Review of Biochemistry*, **68**(1), 649–686.

Van der Vaart, A. W. (2000). *Asymptotic Statistics*, volume 3. Cambridge University Press.

Wang, L., Tang, H., Thayanithy, V., Subramanian, S., Oberg, L., Cunningham, J. M., Cerhan, J. R., Steer, C. J., and Thibodeau, S. N. (2009). Gene networks and microRNAs implicated in aggressive prostate cancer. *Cancer research*, **69**(24), 9490–9497.

Zellner, A. (1986). On assessing prior distributions and bayesian regression analysis with g-prior distributions. *Bayesian inference and decision techniques: essays in honor of Bruno De Finetti*, **6**, 233–243.

Zwiener, I., Frisch, B., and Binder, H. (2014). Transforming RNA-seq data to improve the performance of prognostic gene signatures. *PloS one*, **9**(1), e85150.